



# Affine invariant image comparison

Mariano Rodríguez

## ► To cite this version:

Mariano Rodríguez. Affine invariant image comparison. Computer Vision and Pattern Recognition [cs.CV]. Ecole normale supérieure Paris-Saclay, 2020. English. NNT: . tel-02954027v1

**HAL Id: tel-02954027**

**<https://theses.hal.science/tel-02954027v1>**

Submitted on 30 Sep 2020 (v1), last revised 6 Nov 2020 (v2)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Affine invariant image comparison

**Thèse de doctorat de l'Université Paris-Saclay**

École doctorale n° 574, mathématiques Hadamard (EDMH)  
Spécialité de doctorat: Mathématiques appliquées  
Unité de recherche: Université Paris-Saclay, CNRS, ENS Paris-Saclay,  
Centre Borelli, 91190, Gif-sur-Yvette, France  
Réfèrent: ENS Paris-Saclay

**Thèse présentée et soutenue à Paris, le 10 Juillet 2020, par**

**Mariano RODRIGUEZ**

## Composition du jury:

<b>Coloma Ballester</b> Professeur, Universitat Pompeu Fabra	Rapportrice
<b>Vincent Lepetit</b> Professeur, École des Ponts ParisTech	Rapporteur, Président
<b>Jiri Matas</b> Professeur, Czech Technical University	Rapporteur
<b>Gabriele Facciolo</b> Professeur, École Normale Supérieure Paris-Saclay	Examineur
<b>Patrick Perez</b> Directeur de valeo.ai	Examineur
<b>Julien Rabin</b> Maître de conférences, Université de Caen Normandie	Examineur
<b>Julie Delon</b> Professeur, Université Paris Descartes	Codirectrice
<b>Jean-Michel Morel</b> Professeur, École Normale Supérieure Paris-Saclay	Directeur



**Titre:** Comparaison d'images invariantes affines

**Mots clés:** Invariance affine, mise en correspondance d'images, espace d'inclinaisons, descripteurs locaux, IMAS

**Résumé:** La mise en correspondance d'images, qui consiste à décider si plusieurs images représentent ou non des objets communs ou similaires, est un problème reconnu comme difficile, notamment en raison des changements de point de vue entre les images. Les déformations apparentes des objets causées par les changements de position de la caméra peuvent être approximées localement par des transformations affines. Cette propriété a motivé la recherche de descripteurs locaux invariants affines depuis une quinzaine d'années. Malheureusement, les descripteurs existants ne permettent pas de traiter des différences de point de vue d'angle supérieures à 45 degrés, et échouent complètement au-delà de 60 degrés. Dans cette thèse, nous abordons plusieurs stratégies pour résoudre cette limitation, et nous montrons qu'elles se complètent.

**Title:** Affine invariant image comparison

**Keywords:** affine invariance, image matching, space of tilts, local descriptors, IMAS

**Abstract:** Image comparison, which consists in deciding whether or not several images represent some common or similar objects, is a problem recognized as difficult, especially because of the viewpoint changes between images. The apparent deformations of objects caused by changes of the camera position can be locally approximated by affine maps. This has motivated the quest for affine invariant local descriptors in the last 15 years. Unfortunately, existing descriptors cannot handle angle viewpoint differences larger than 45 degrees, and fail completely beyond 60 degrees. In this thesis, we address several strategies to resolve this limitation, and we show at the end that they complete each other.

*a mis padres*



## Acknowledgments

To be completely honest I must admit that many stars have lined up for me to be here today. In addition, life has given me the chance of working on a field I enjoy. Therefore, I would like to start these lines of acknowledgment by thanking that invisible helping hand who has arranged a series of events in my favour.

I am extremely grateful to my supervisors, Julie Delon and Jean-Michel Morel, for their guidance and support, both professionally and personally, during these exciting four years of PhD studies. They have undeniably succeeded in passing on to me their love for research. Many thanks to my collaborators, and nowadays also friends, Gabriele Facciolo, Rafael Grompone von Gioi and Pablo Musé, with whom I took great pleasure to work with. I am also thankful to the entire lab, i.e., PhD students, postdocs, secretaries and professors, for providing such a healthy and friendly atmosphere.

I would like to thank the FSMP (Fondation Sciences Mathématiques de Paris) for giving me the opportunity in 2014 to come and study in such a specialist environment with leading academics in this field. Many thanks to the french people in general for sharing what nowadays has also become my culture. I cannot hide my pride today when they call me: '*cher compatriote*'. Of course, I have to mention those groups of friends that have provided me with stability through these six years in France. I am thinking of the Italian group, the first group to adopt me and from where I met one of my best friends ever, Giulia; the Uruguayan group, the second group to adopt me, a very functional group with friends that should last a lifetime; and, this very last year, the Cuban group, where I enjoy the company of extraordinary people and the flavour of my roots.

I would like to finish by thanking my dear family. Special thanks to my parents, Gilda and Mariano, for a lifetime of love, guidance, sacrifice, dedication and constant source of inspiration; to tía Tere and tío Stephen for being the ignition and guardian angels of this adventure; to tía Stella, tío Nestor, tía Laura, tía Laurita, and primos Sasha and Jammal for being always there to help me in times of difficulty; to uncle George and oncle Dominique, my Parisian family, for providing shelter, support and wonderful '*apéros*'. Thanks to my sister Gilma, my emotional lifter, and to tío Gilberto and tío Tico for fruitful discussions. I also want to thank the rest of the family for providing me with all the love and strength needed to face the unknown of this journey.

Thank you very much to all of you !

Mariano Rodríguez Guerra



## Abstract

Image comparison, which consists in deciding whether or not several images represent some common or similar objects, is a problem recognized as difficult, especially because of the viewpoint changes between images. The apparent deformations of objects caused by changes of the camera position can be locally approximated by affine maps. This has motivated the quest for affine invariant local descriptors in the last 15 years. Unfortunately, existing descriptors cannot handle angle viewpoint differences larger than 45 degrees, and fail completely beyond 60 degrees. In this thesis, we address several strategies to resolve this limitation, and we show at the end that they complete each other.

Three main branches to obtain affine invariance are actively being investigated by the scientific community:

1. Through affine simulations followed by (less invariant) matching of many simulated image pairs;
2. Through a description that is already independent from the viewpoint;
3. Through local affine patch normalization.

In this thesis we explore all three approaches. We start by presenting a distance between affine maps that measures viewpoint deformation. This distance is used to generate optimal (minimal) sets of affine transformations, to be used by Image Matching by Affine Simulation (IMAS) methods. The goal is to reduce the number of affine simulations while keeping the same performance level in the matching process. We use these optimal sets of affine maps and other computational improvements to boost the well established ASIFT method. We also propose a new method, Optimal Affine-RootSIFT whose performance and speed significantly improve on those of ASIFT. As a side quest and direct application of the IMAS methodology, we propose two descriptors suitable to track repeated objects based on the Number of False Alarms (NFA), test their viewpoint tolerance and generate accordingly proper sets of affine simulations. In that way we end up with two IMAS methods able to handle repetitive structures under strong viewpoint differences.

Our search for improvement focuses then on local descriptors, which once were manually-designed, but are currently being learned from data with the promise of a better performance. This motivates our proposition of an affine invariant descriptor (called AID) based on a convolutional neural network trained with optical affine simulated data. Even if not trained for occlusion nor noise, the performance of AIDs on real images is surprisingly good. This performance confirms that it might be possible to attain a straightaway common description of a scene regardless of viewpoint.

Finally, recent advances in affine patch normalization (e.g. Affnet) help circumvent the lack of affine invariance of state-of-the-art descriptors. As usual with affine normalization, patches are normalized to a single representation and then described. We instead propose to rely not on the precision nor on the existence of a single affine normalizing map, by presenting an Adaptive IMAS method that computes a small set of possible normalizing representations. This method aggregates the Affnet information to attain a good compromise between speed and performance. At the end of the day, our inquiries lead to a method that fuses normalization and simulation ideas to get a still faster and more complete affine invariant image matcher.

All in all, affine invariance is a way to remove the viewpoint information from patches and focus on what the scene really describes. However, clues on how geometry is transformed can be useful when matching two images, e.g., recovering the global transformation, the proposal of new tentative matches, among others. With that in mind, we

propose a LOCal Affine Transform Estimator (LOCATE) which is proved to be valuable for affine guided matching and homography estimation. These two applications of LOCATE provide complementary tools that improve still more the affine invariant image matchers presented above.

## Résumé

La mise en correspondance d’images, qui consiste à décider si plusieurs images représentent ou non des objets communs ou similaires, est un problème reconnu comme difficile, notamment en raison des changements de point de vue entre les images. Les déformations apparentes des objets causées par les changements de position de la caméra peuvent être approximées localement par des transformations affines. Cette propriété a motivé la recherche de descripteurs locaux invariants affines depuis une quinzaine d’années. Malheureusement, les descripteurs existants ne permettent pas de traiter des différences de point de vue d’angle supérieures à 45 degrés, et échouent complètement au-delà de 60 degrés. Dans cette thèse, nous abordons plusieurs stratégies pour résoudre cette limitation, et nous montrons qu’elles se complètent.

Trois directions principales pour obtenir l’invariance affine sont activement étudiées par la communauté scientifique :

1. Par des simulations affines suivies d’un appariement (moins invariant) de nombreux couples d’images simulées ;
2. Par une description indépendante du point de vue ;
3. Grâce à une normalisation affine locale de patches.

Dans cette thèse, nous explorons les trois approches. Nous commençons par présenter une distance entre les transformations affines qui mesure la déformation du point de vue. Cette distance est utilisée pour générer des ensembles optimaux (minimaux) de transformations affines, qui sont utilisés par les méthodes de mise en correspondance d’images par simulation affine (IMAS). L’objectif est de réduire le nombre de simulations affines à simuler tout en conservant le même niveau de performance dans le processus d’appariement. Nous utilisons ces ensembles optimaux de transformations affines et d’autres améliorations informatiques pour renforcer la méthode ASIFT. Nous proposons également une nouvelle méthode, Optimal Affine-RootSIFT, dont les performances et la vitesse sont nettement supérieures à celles d’ASIFT. Dans une application directe de la méthodologie IMAS pour un problème connexe, nous proposons deux descripteurs permettant de suivre des objets répétés en mesurant un nombre de fausses alarmes (NFA), de tester leur tolérance au changement de point de vue, et de générer en conséquence des ensembles appropriés de simulations affines. De cette façon, nous obtenons deux méthodes IMAS capables de traiter des structures répétitives avec de fortes différences de points de vue.

Notre recherche d’amélioration se concentre ensuite sur les descripteurs locaux, qui étaient autrefois conçus heuristiquement, mais qui sont actuellement appris à partir de données massives, avec la promesse d’une meilleure performance. Nous proposons un descripteur invariant affine (appelé AID) appris par un réseau neuronal convolutionnel entraîné avec des données optiques affines simulées. Même si ce réseau n’est pas entraîné pour les occlusions ou le bruit, la performance des descripteurs AIDs sur des images réelles est étonnamment bonne. Cette performance confirme qu’il est possible d’obtenir immédiatement une description commune d’une scène, quel que soit le point de vue.

Enfin, les progrès récents dans la normalisation affine des patches (par exemple Affnet) permettent de contourner l’absence d’invariance affine des descripteurs de l’état de l’art. Comme d’habitude avec la normalisation affine, les patches sont normalisés en une représentation unique, qui est transformée en descripteur. Nous préférons ne pas nous fier à la précision ni à l’existence d’une seule normalisation affine, et présentons une méthode



IMAS adaptative qui calcule un petit ensemble de représentations normalisantes possibles. Cette méthode agrège les informations d’Affnet pour obtenir un bon compromis entre vitesse et performance. En fin de compte, nos recherches aboutissent à une méthode qui fusionne les idées de normalisation et de simulation pour obtenir une mise en correspondance d’images invariante affine encore plus rapide et plus complète.

# Contents

<b>Acknowledgments</b>	<b>5</b>
<b>Abstract</b>	<b>7</b>
<b>Résumé</b>	<b>9</b>
<b>Introduction</b>	<b>13</b>
<b>Introduction en français</b>	<b>25</b>
 <b>I Image Matching by Affine Simulations</b>	 <b>37</b>
<b>1 Covering the space of tilts</b>	<b>39</b>
1.1 Introduction . . . . .	39
1.2 The space of affine tilts . . . . .	42
1.3 Application: optimal affine invariant image matching algorithms . . . . .	52
1.4 Experimental Validation . . . . .	61
<b>2 Fast affine invariant image matching</b>	<b>71</b>
2.1 Introduction . . . . .	71
2.2 Image Matching by Affine Simulation . . . . .	72
2.3 Hyper-Descriptors in IMAS . . . . .	80
2.4 Two Structural and Computational Improvements . . . . .	85
2.5 Numerical Results . . . . .	86
<b>3 Affine invariant image comparison under repetitive structures</b>	<b>95</b>
3.1 Introduction . . . . .	95
3.2 The gradient angle field descriptor . . . . .	96
3.3 A contrario match validation . . . . .	96
3.4 Affine invariance . . . . .	98
3.5 Experiments . . . . .	99
<b>4 Conclusion</b>	<b>105</b>
 <b>II Learning the affine world</b>	 <b>109</b>
<b>5 AID: an Affine Invariant Descriptor</b>	<b>111</b>

5.1	Introduction . . . . .	111
5.2	Affine viewpoint simulation . . . . .	112
5.3	Descriptors and matching criteria . . . . .	113
5.4	Experiments . . . . .	115
<b>6</b>	<b>Robust estimation of local affine maps</b>	<b>119</b>
6.1	Introduction . . . . .	119
6.2	Affine Maps and Homographies . . . . .	120
6.3	The Local Affine Transform Estimator . . . . .	123
6.4	Refinement and Guided Matching . . . . .	125
6.5	Robust Homography Estimation . . . . .	126
6.6	Experiments . . . . .	129
<b>7</b>	<b>CNN-assisted coverings in the Space of Tilts</b>	<b>133</b>
7.1	Introduction . . . . .	133
7.2	Affine maps and the space of tilts . . . . .	135
7.3	Adaptive coverings . . . . .	136
7.4	Experiments . . . . .	138
<b>8</b>	<b>Conclusion</b>	<b>143</b>
	<b>Appendices</b>	<b>151</b>
<b>A</b>	<b>Proof of Theorem 1.1</b>	<b>151</b>
	<b>Bibliography</b>	<b>155</b>

# Introduction

Image matching aims at establishing correspondences between similar objects that appear in different images. This is a fundamental step in many computer vision and image processing applications such as scene recognition [VGS10, BS11, SAS07, FBA<sup>+</sup>06, MP04, RdSLD07, VvHR05, GL06, YC07, MP07] and detection [FSKP, NTG<sup>+</sup>06], object tracking [ZYS09], robot localization [SLL01, VL10, MMK06, BSBB06, NS06], image stitching [AAC<sup>+</sup>06, BL03], image registration [YSST07, LYT11] and retrieval [HL04, GLGP13], 3D modeling and reconstruction [Fau93, GZS11, VV05, AFS<sup>+</sup>11, RTA06], motion estimation [WRHS13], photo management [SSS06, Yan07, LCC06, Cha05], symmetry detection [LE06] or even image forgeries detection [CPV15].

The general (solid) shape matching problem starts with several photographs of a physical object, possibly taken with different cameras and viewpoints. These digital images are the *query* images. Given other digital images, the *target* images, the question is whether some of them contain, or not, a view of the object taken in the query image. This problem is by far more restrictive than the *categorization* problem, where the question is to recognize a *class* of objects, like chairs or cats. In the shape matching framework several instances of the very *same* object, or of copies of this object, are to be recognized. The difficulty is that the change of camera position induces an apparent deformation of the object's image. Thus, recognition must be invariant with respect to such deformations. Let us point out that this matching problem consists in localizing common structures between images, but also in deciding whether a structure is present or not in an image. Indeed, computer vision systems have to deal with situations where the object of interest is not present. It is thus of great interest to limit the number of false matches, especially in the case of very large image databases.

The state-of-the-art image matching algorithms usually consist of three parts: *detector*, *descriptor* and *matching step*. They first detect points of interest in the compared images and select a region around each point of interest, and then associate an invariant descriptor or feature to each region. Correspondences may thus be established by matching the descriptors. Detectors and descriptors should be as invariant as possible.

Local image detectors can be classified by their incremental invariance properties. All of them are translation invariant. The Harris point detector [HS88] is also rotation invariant. The Harris-Laplace, Hessian-Laplace and the DoG (Difference-of-Gaussian) region detectors [MS01, MS04, Low04, F  v07] are invariant to rotations and changes of scale. Based on the AGAST [MHB<sup>+</sup>10] corner score, BRISK [LCS11] performs a 3D nonmaxima suppression and a series of quadratic interpolations to extract the BRISK keypoints; both detections aim to quickly provide rotation and scale invariances. Some moment-based region detectors [LG94, Bau00] including the Harris-Affine and Hessian-Affine region detectors [MS02, MS04], an edge-based region detector [TVO99, TV04], an intensity-based region detector [TV00, TV04], an entropy-based region detector [KZB04], and two level

line-based region detectors MSER (“maximally stable extremal region”) [MCUP04] and LLD (“level line descriptor”) [MSCG03, MSC<sup>+</sup>06, CLM<sup>+</sup>08] are designed to be invariant to affine transforms. MSER, in particular, has been demonstrated to have often better performance than other affine invariant detectors, followed by Hessian-Affine and Harris-Affine [MTS<sup>+</sup>05].

In his keystone paper [Low04], Lowe has proposed a scale-invariant feature transform (SIFT) that is invariant to image scaling and rotation and partially invariant to illumination and viewpoint changes. The SIFT method combines the DoG region detector that is rotation, translation and scale invariant (a mathematical proof of its scale invariance is given in [MY08]) with a descriptor based on the gradient orientation distribution in the region, which is partially illumination and viewpoint invariant [Low04]. These two stages of the SIFT method will be called respectively *SIFT detector* and *SIFT descriptor*. The SIFT detector is *a priori* less invariant to affine transforms than the Hessian-Affine and the Harris-Affine detectors [MS01, MS04].

The SIFT descriptor has been shown to be superior to other many descriptors [MS05] such as the distribution-based shape context [ATRB95], the geometric histogram [ATRB95] descriptors, the derivative-based complex filters [Bau00, SZ02], and the moment invariants [VMU96]. A number of SIFT descriptor variants and extensions, including PCA-SIFT [KS04], GLOH (gradient location-orientation histogram) [MS05] and SURF (speeded up robust features) [BTV06] have been developed ever since [FS07, LÁJA06]. More recently, RootSIFT has been proposed in [AZ12], which suggests a slight modification to SIFT descriptors in order for them to be compared by a Hellinger kernel. Most of these SIFT variants claim more robustness and distinctiveness with scaled-down complexity.

Binary descriptors constitute another branch of local descriptors that focuses on speed and efficiency. In BRIEF [CLSF10] a smoothed local patch surrounding the keypoint is sampled in a set of random pixel pairs whose values are compared, thus producing a binary string. This allows to exploit differential characteristics of the local patch in a different way from the gradient histogram-based approaches like SIFT, and leads to a compact signature, represented by a sequence of bits. Similarly, the BRISK descriptor is a binary string resulting from brightness differences computed around the keypoint. Another alternative to binary local image descriptors is LATCH [LH16] which is a fast and compact binary descriptor used to represent local image regions. Its authors claim that its extraction time is only slightly longer than other binary descriptors and far faster than floating point representations.

Nonlinear scale spaces have also been used in order to make blurring locally adaptive to the image data; blurring small details but preserving object boundaries. The KAZE features method, introduced in [ABD12], claims that it increases repeatability and distinctiveness with respect to SIFT and SURF thanks to the use of a nonlinear diffusion filter. The main drawback of KAZE is speed. An accelerated version of KAZE, called AKAZE [ANB13], generates computationally less demanding features while still based on nonlinear diffusion.

The mentioned state-of-the-art methods have achieved brilliant success. However, none of them is fully affine invariant. As pointed out in [Low04], Harris-Affine and Hessian-Affine start with initial feature scales and locations selected in a non-affine invariant manner. The non-commutation of optical blur and affine transforms pointed out in [YM11] explains the limited affine invariance performance of the normalization methods MSER, LLD, Harris-Affine and Hessian-Affine. As shown in [CLM<sup>+</sup>08], MSER and LLD are not scale invariant: they do not cope with the drastic level line structure changes caused by blur. SIFT (and its direct variants) is actually the only method that

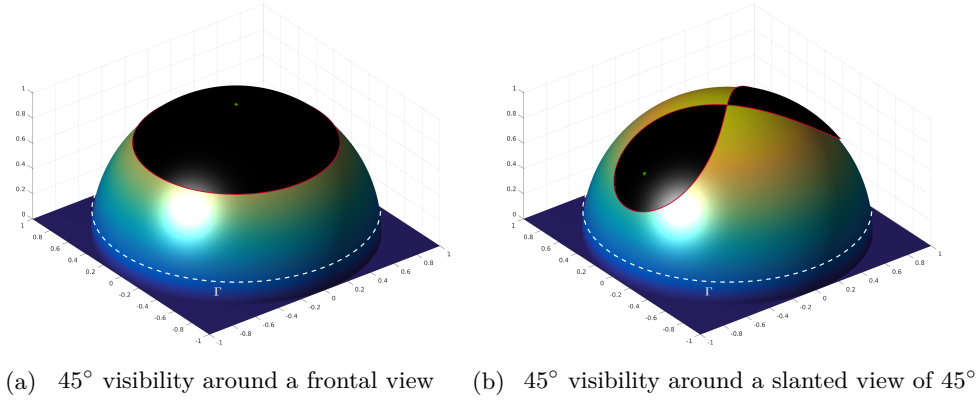


Figure 1: (Perspective views)

Green point - Query camera position  
Dashed line -  $\partial B([Id], \log 4\sqrt{2})$   
Black surface - Visible target camera positions

is fully scale invariant. However, since it is not designed to cover the whole affine space, its performance drops quickly under substantial viewpoint changes. In fact, all the aforementioned descriptors are not fully invariant to viewpoint changes; they have been proved to be recognizable under viewpoint angles up to  $60^\circ$  for planar objects [YM11, MMP15] but with a drastic performance drop for angles larger than  $45^\circ$  [Kar16]. In Figure 1, we show fixed query camera positions (green points) and all possible target camera positions (black surfaces) with affine viewpoints under  $45^\circ$ .

In this thesis the question of viewpoint invariance in image matching is central. As many others in the literature, we reach this objective through affine invariance, which can be obtained by exploiting affine properties at different places of the matching pipeline. Our main contributions regarding image matching under strong viewpoint changes are shown in Table 1. We list in Table 2 several tools and methods designed to validate our claims. Table 3 presents all techniques appearing in this thesis to attain affine invariance, and ultimately viewpoint invariance. Finally, Table 4 enumerates all affine invariant matching methods appearing in this thesis.

## Covering the Space of Tilts

In Chapter 1 we start by computing exact formulas to determine those black accessibility surfaces displayed in Figure 1. This amounts to obtain a formula for the distance that measures tilt deformation. We then propose a mathematical method to analyze the numerous algorithms performing Image Matching by Affine Simulation (IMAS). To become affine invariant they apply a discrete set of affine transforms to the images, previous to the comparison of all images by a Scale Invariant Image Matching (SIIM) like SIFT, see Figure 2. Obviously, this multiplication of images to be compared increases the image matching complexity. Three questions arise: a) what is the best set of affine transforms to apply to each image to gain full practical affine invariance? b) what is the lowest attainable complexity for the resulting method? c) how to choose the underlying SIIM method? We provide an explicit answer and a mathematical proof of quasi-optimality of the solution to the first question. Figure 3 highlights the benefits of this methodology by optimizing the set of affine simulations of the classic ASIFT [MY09, YM11] method. As an answer to b) we find that the near-optimal complexity ratio between full affine matching and scale invariant matching is more than halved, compared to the current

**Table 1** Main contributions to image comparison under strong viewpoint changes.

Proposals	Chap. 1	Chap. 2	Chap. 3	Chap. 5	Chap. 6	Chap. 7
Analytic formulas to measure tilt deformations	✓					
Optimal coverings in the Space of Tilts	✓	✓				✓
Distinctive matchers able to capture repetitive structures under viewpoint		✓	✓	✓		
Viewpoint invariant descriptors beyond 60°				✓		
Estimating local geometry transformations					✓	
Complexity reduction	✓	✓		✓		✓

**Table 2** Tools and methods for validation of affine properties.

Validation	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
Synthesized data	✓		✓		✓	✓		✓
Successful homography retrievals on real images	✓	✓	✓	✓	✓	✓	✓	✓
Successful homography retrievals on more than one database				✓		✓	✓	✓
Density estimations on descriptors measurements					✓			✓
Density estimations of affine parameters						✓		
Density estimations in the Space of Tilts							✓	
ROC curves								✓

**Table 3** Techniques to attain affine invariance.

Techniques	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
Optical affine simulations	✓	✓	✓	✓	✓		✓	✓
Direct descriptions					✓	✓	✓	✓
Patch normalization						✓	✓	✓

**Table 4** Available methods for affine invariant image comparison. No reference indicated means that the method has been implemented in this thesis. Appearing in braces, all possible descriptors for a fixed setting.

Methods	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
ASIFT [MY09]	✓			✓			✓	
FAIR-SURF [PLYP12]	✓	✓						
Optimal Affine-RootSIFT	✓	✓	✓	✓	✓		✓	
Optimal Affine-RootSIFT Revisited		✓		✓				
Optimal Affine-SURF	✓	✓		✓				
Optimal Affine-{SIFT, BRISK, BRIEF, ORB, AKAZE, LATCH, FREAK, AGAST, LUCID, DAISY}	✓	✓						
Optimal Affine-{LDA64, LDA128, DIF64, DIF128, HalfRootSIFT, HalfSIFT}		✓						
Optimal Affine-{AC, AC-W, AC-Q}			✓	✓				
SIFT-AID					✓	✓	✓	
SIFT+Affnet+HardNet						✓		
HesAffNet [MRM18]							✓	✓
Adaptive-ARootSIFT, Greedy-ARootSIFT							✓	
HesAff-{AID, AID21}								✓
HesAffnet-{AID, AID21}								✓



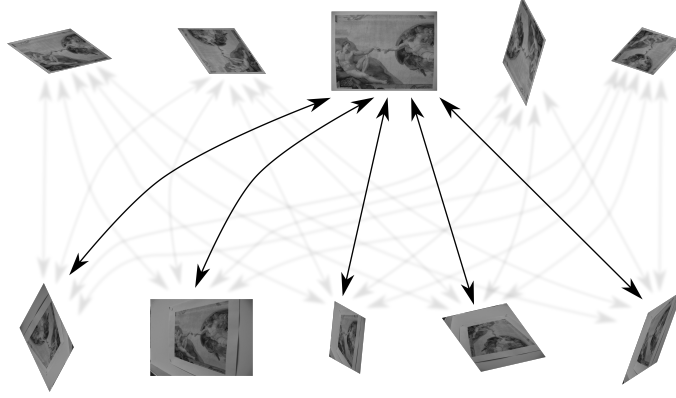
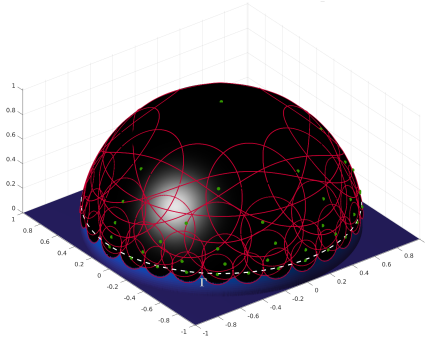
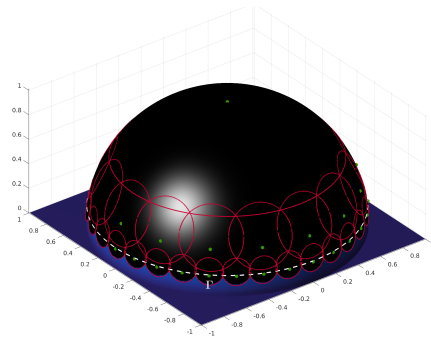


Figure 2: IMAS algorithms start by applying a finite set of optical affine simulations to  $u$  and  $v$ , followed by pairwise comparisons.



(a) ASIFT [MY09, YM11], a  $56^\circ$ -covering with a set of 41 affine simulations that renders target camera viewpoints visible up to  $80^\circ$ . Highly redundant, nonetheless it does cover the region it was meant to.



(b) Our proposition,  $56^\circ$ -covering with a set of 28 affine simulations that renders target camera viewpoints visible up to  $82^\circ$ .

Figure 3: Coverings

- Green points - Affine camera simulations
- Red lines - Visibility tolerance from each affine simulation
- Black surfaces - Visible viewpoints regions
- Dashed line - Covered regions

IMAS methods. This means that the number of key points necessary for affine matching can be halved, and that the matching complexity is divided by four for exactly the same performance. This also means that an affine invariant set of descriptors can be associated with any image. The price to pay for full affine invariance is that the cardinality of this set is around 6.4 times larger than for a SIIM.

## Fast affine invariant image matching

Chapter 2 focuses mainly on implementation details, two structural modifications and some other improvements to IMAS methods. As stated before, these methods attain affine invariance by applying a finite set of affine transforms to the images before comparing them with a SIIM method like SIFT or SURF. We describe in Chapter 2 how to optimize these IMAS methods. First, we detail an algorithm computing a minimal discrete set of

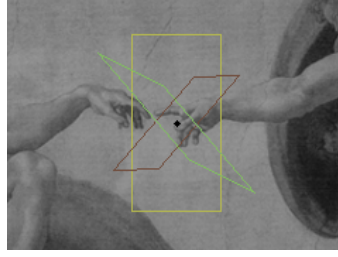


Figure 4: A hyper-descriptor  $d = \{\omega_1, \omega_2, \omega_3\}$

Black point - Center of  $d$  and of each  $\omega_i$   
 Colored parallelograms - Affine views described by  $\omega_i$   
 (Affine views are square patches of regions enclosed by parallelograms)

affine transforms to be applied to each image before comparison. It yields a full practical affine invariance at the lowest computational cost. The matching complexity of current IMAS algorithms is divided by about 4. Our approach also associates to each image an affine invariant set of descriptors, which is twice smaller than the set of descriptors usually used in IMAS methods, and only 6.4 times larger than the set of similarity invariant descriptors of SIIM methods. In order to reduce the number of false matches, which are inherently more frequent in IMAS approaches than in SIIM, we introduce the notion of hyper-descriptor, which groups descriptors whose keypoints are spatially close. Hyper-descriptors aim to provide different affine representations of a common scene. Figure 4 shows three affine zones that are to be described and grouped into a hyper-descriptor. Finally, we also propose a matching criterion allowing each keypoint of the query image to be matched with several keypoints of the target image, in order to deal with situations where an object is repeated several times in the target image. An online demo allowing to reproduce all results is available in the IPOL article

<https://ipolcore.ipol.im/demo/clientApp/demo.html?id=225>,

where the source code is also made available. Compilation and usage instructions are included in the `README.md` file of the archive. Complementary information on Chapter 2 is available at the web page of this work:

<https://rdguez-mariano.github.io/pages/hyperdescriptors>.

## Image comparison under repetitive structures

In Chapter 3 we focus on the problem of affine invariant image comparison in the presence of noise and repetitive structures. The classic scheme of keypoints, descriptors and matcher is used. A local field of image gradient orientations is used as descriptor (see Figure 5) and two matchers are proposed, based on the a-contrario theory, for handling repetitive structures. The affine invariance is obtained by affine simulations. The proposed methods achieve state-of-the-art performance under repetitive structures. Complementary information on Chapter 3 is available at the web page of this work:

<https://rdguez-mariano.github.io/pages/acdesc>.

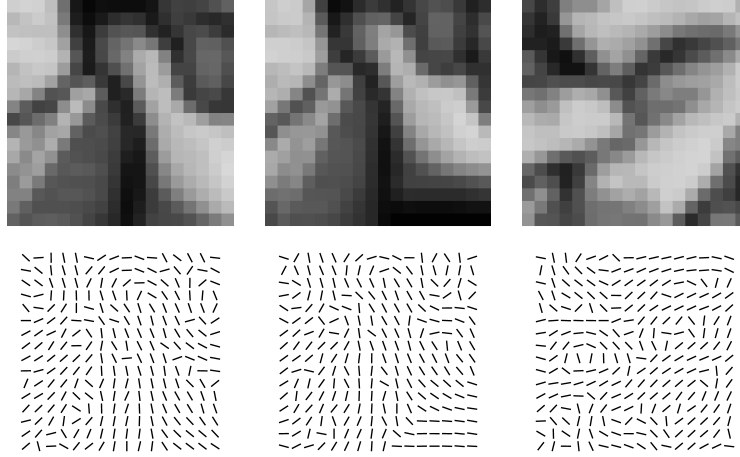


Figure 5: Three image patches and their corresponding orientation fields used as descriptors. The first two are similar while the third one is different.

## AID: An affine invariant descriptor

In the same line as in previous chapters, the classic approaches to image descriptors are adapted in Chapter 5. A descriptor encodes the local information around the keypoint. An advantage of local approaches is that viewpoint deformations are well approximated by affine maps. This has motivated the never ending quest for affine invariant local descriptors. Despite numerous efforts, such descriptors have remained elusive, ultimately resulting in a compromise between using viewpoint simulations or patch normalization to attain affine invariance. In Chapter 5 we propose a CNN-based patch descriptor which captures affine invariance without the need for viewpoint simulations or patch normalization. This is achieved by training a neural network to associate similar vectorial representations to patches related by affine transformations. During matching, these vectors are compared very efficiently. The method’s matching invariance to translation, rotation and scale is still obtained by the first stages of SIFT, which produce the keypoints. The proposed descriptor outperforms the state-of-the-art descriptors in retaining affine invariant properties. Figure 6 shows density estimations for positive and negative patch pairs for RootSIFT, BigAID (ours) and AID (ours). Complementary information on Chapter 5 is available at the web page of this work:

<https://rdguez-mariano.github.io/pages/siftaid>.

## Robust estimation of local affine maps

The corresponding point coordinates determined by classic image matching approaches define local zero-order approximations of the global mapping between two images. But the patches around keypoints typically contain more information, which may be exploited to obtain a first-order approximation of the mapping, incorporating local affine maps between corresponding keypoints. See Figure 7 for a visual representation of these first-order approximations. In Chapter 6, we propose a LOCal Affine Transform Estimator (LOCATE) method learned by a neural network. We show that LOCATE drastically improves the accuracy of local geometry estimation by tracking inverse maps. A second contribution on

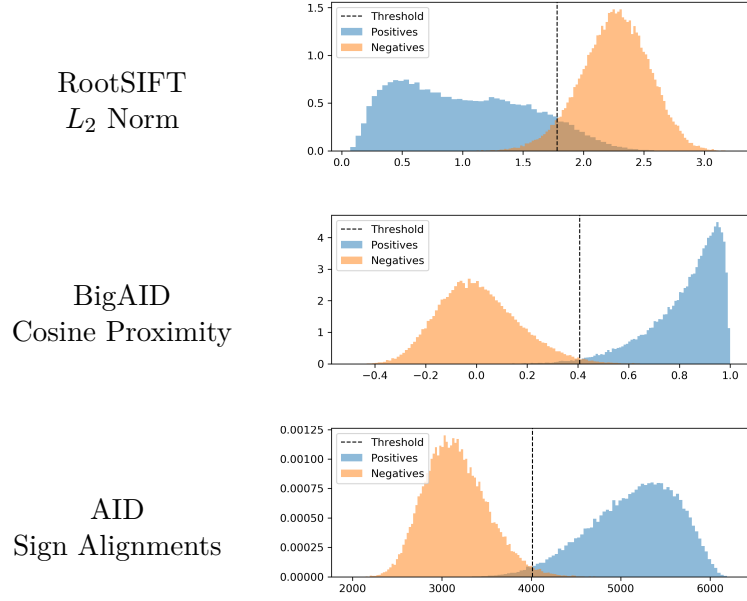


Figure 6: Positive and negative density estimation on measurements. For that,  $6 \cdot 10^5$  random intra and extra class pairs were used. The vertical line depicts the threshold minimizing both error probabilities: false negatives and false positives.

guided matching and refinement is also presented. The novelty here consists in the use of LOCATE to propose new SIFT-keypoint correspondences with precise locations, orientations and scales. Our experiments show that the precision gain provided by LOCATE does play an important role in applications such as guided matching. The third contribution of this chapter consists in a modification of the RANSAC algorithm, that uses LOCATE to improve the homography estimation between a pair of images. These approaches outperform RANSAC for different choices of image descriptors and image datasets, and permit to increase the probability of success in identifying image pairs in challenging matching databases. Complementary information on Chapter 3 is available at the web page of this work:

<https://rdguez-mariano.github.io/pages/locate>.

## CNN-assisted coverings in the Space of Tilts

As stated above, affine invariant descriptors have remained elusive, which explains the development of IMAS methods. These methods simulate viewpoint changes to attain the desired invariance. Yet, recent CNN-based methods seem to provide a way to learn affine invariant descriptors. Still, as a first contribution, we show that current CNN-based methods remain far from reaching the state-of-the-art performance provided by IMAS. This confirms that there is still room for improvement for learned methods. Second, we show that recent advances in affine patch normalization can be used to create adaptive IMAS methods that select their affine simulations depending on query and target images. Figure 8 shows kernel density estimations in the Space of Tilts (formally introduced in Chapter 1) for query and target images in the ‘cat’ pair from the EVD [MMP15] dataset. Notice the concentration around orthogonal directions in the Space of Tilts of affine maps provided by Affnet [MRM18] from query and target images. Just by looking at those

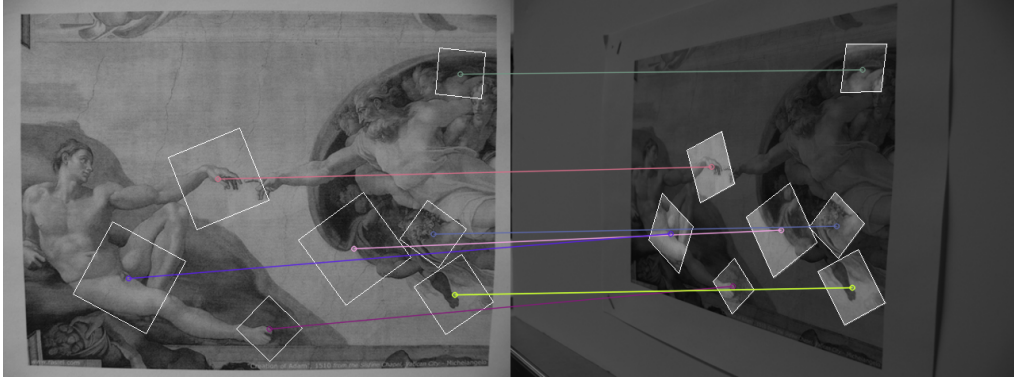


Figure 7: Some correspondences together with local affine maps estimated by the proposed LOCATE network. Patches on the target are warped versions of their corresponding query patch.

densities one can already infer that the common object to both images was seen from camera positions that differ by  $90^\circ$ . In practice, Affnet [MRM18] predictions will be used to select convenient affine transformations to be tested in IMAS methods. The proposed methods are shown to attain a good compromise: on the one hand, they reach the performance of state-of-the-art IMAS methods but are faster; on the other hand, they perform significantly better than non-simulating methods, including recent ones. Complementary information on Chapter 3 is available at the web page of this work:

<https://rdguez-mariano.github.io/pages/adimas>.

## Published work

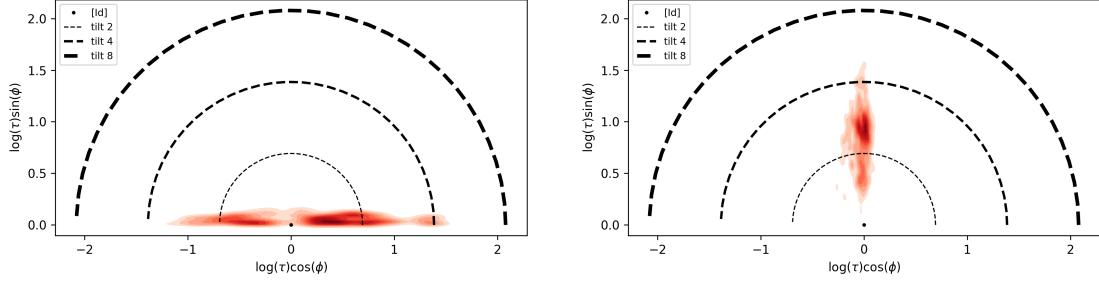
The research papers linked to this thesis are:

1. [RDM18a]. M. Rodriguez, J. Delon, and J.-M. Morel. Covering the space of tilts: application to affine invariant image comparison. In *SIIMS*, 11(2):1230–1267, 2018.
2. [RDM18b]. M. Rodriguez, J. Delon, and J.-M. Morel. Fast affine invariant image matching. In *IPOLE*, 8:251–281, 2018.
3. [RGvG18]. M. Rodriguez and R. Grompone von Gioi. Affine invariant image comparison under repetitive structures. In *ICIP*, pages 1203–1207, Oct 2018.
4. [RFGvG<sup>+</sup>19]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, J.-M. Morel, and J. Delon. SIFT-AID: boosting SIFT with an affine invariant descriptor based on convolutional neural networks. In *ICIP*, Sep 2019.
5. [RFGvG<sup>+</sup>20]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, and J. Delon. Robust estimation of local affine maps and its applications to image matching. In *WACV*, 2020.
6. [RFvG<sup>+</sup>20]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, J. Delon, and J.-M. Morel. CNN-assisted coverings in the space of tilts: best affine invariant performances with the speed of CNNs. In *ICIP*, Oct 2020.

Source codes have been published in github. They are:



(a) Common object to query (left) and target (right) images.



(b) Kernel density estimations of query (left) and target (right) Affnet [MRM18] affine maps.

Figure 8: Kernel density estimations in the Space of Tilts of affine maps extracted by Affnet [MRM18] for both images in the 'cat' pair from the EVD [MMP15] dataset.

1. The tester framework, figures and the covering optimizer appearing in [RDM18a] can be found at:

[https://github.com/rdguez-mariano/imas\\_analytics](https://github.com/rdguez-mariano/imas_analytics)

2. All IMAS methods presented in [RDM18b, RGvG18] can be found at:

[https://github.com/rdguez-mariano/fast\\_imas\\_IPOL](https://github.com/rdguez-mariano/fast_imas_IPOL)

3. The SIFT-AID method as well as all results appearing in [RFGvG<sup>+</sup>19] can be found at:

<https://github.com/rdguez-mariano/sift-aid>

4. The LOCATE method, plus two applications (affine guided matching and a version of RANSAC homography), some other matching methods (e.g. HessAff+Affnet+HardNet, HessAff+AID and SIFT+Affnet+HardNet) and all results appearing in [RFGvG<sup>+</sup>20] can be found at:

<https://github.com/rdguez-mariano/locate>

5. The Adaptive IMAS methods appearing in [RFvG<sup>+</sup>20] can be found at:

[https://github.com/rdguez-mariano/fast\\_imas\\_IPOL/tree/master/adaptiveIMAS](https://github.com/rdguez-mariano/fast_imas_IPOL/tree/master/adaptiveIMAS)



# Introduction en français

La comparaison d'images vise à établir des correspondances entre des objets similaires qui apparaissent dans différentes images. Il s'agit d'une étape fondamentale dans de nombreuses applications de vision par ordinateur et de traitement d'images, telles que la reconnaissance de scènes [VGS10,BS11,SAS07,FBA<sup>+</sup>06,MP04,RdSLD07,VvHR05,GL06,YC07,MP07], la détection d'objet [FSKP,NTG<sup>+</sup>06], le suivi d'objet [ZYS09], la localisation de robot [SLL01,VL10,MMK06,BSBB06,NS06], l'assemblage d'images [AAC<sup>+</sup>06,BL03], le recalage d'image [YSST07,LYT11], la recherche par le contenu [HL04,GLGP13], la modélisation et reconstruction 3D [Fau93,GZS11,VV05,AFS<sup>+</sup>11,RTA06], l'estimation de mouvement [WRHS13], la gestion de photos [SSS06,Yan07,LCC06,Cha05], la détection de symétrie [LE06] ou même la détection de contrefaçons [CPV15].

Le problème général de la correspondance de formes (solides) suppose que l'on ait plusieurs photographies d'un objet physique, éventuellement prises avec des caméras et des points de vue différents. Ces images numériques sont les images *requêtes*. Étant donné d'autres images numériques, les images *cibles*, la question est de savoir si certaines parmi elles contiennent ou non une vue de l'objet présent dans l'image requête. Ce problème est bien plus difficile que le problème de la *catégorisation*, où il s'agit de reconnaître une *classe* d'objets, comme des chaises ou des chats. Dans le cadre de la correspondance de formes, plusieurs instances de l'objet même, ou des copies de cet objet, doivent être reconnues. La difficulté est que le changement de position de la caméra induit une déformation apparente de l'objet dans l'image. Ainsi, la reconnaissance doit être invariable aux déformations. Signalons que ce problème de mise en correspondance consiste à localiser des structures communes entre les images, mais aussi à décider si une structure est présente ou non dans une image. En effet, les systèmes de vision par ordinateur doivent faire face à des situations où l'objet d'intérêt n'est pas nécessairement présent. Il est donc important de limiter le nombre de fausses correspondances, notamment dans le cas de très grandes bases d'images.

Les algorithmes les plus performants pour la mise en correspondance d'images sont divisés généralement en trois étapes : *détection*, *description* et *appariement*. Ils détectent d'abord les points d'intérêt dans les images à comparer, sélectionnent une région autour de chaque point d'intérêt, puis ils associent un descripteur, ou une caractéristique invariante, à chaque région. La mise en correspondance consiste alors à appairer ces descripteurs. Les étapes de détection et de description doivent être aussi invariantes que possible.

Les détecteurs de points d'intérêt peuvent être classés selon leurs propriétés d'invariance de manière incrémentale. Tous sont invariants par translation. Le détecteur de point Harris [HS88] est également invariant par rotation. Les détecteurs Harris-Laplace, Hessian-Laplace et le Détecteur de régions DoG (Différence de Gaussiennes) [MS01,MS04,Low04,Fév07] sont invariants aux rotations et aux changements d'échelle. Sur la base du score du détecteur de coin AGAST [MHB<sup>+</sup>10], le détecteur BRISK [LCS11] effectue une sup-



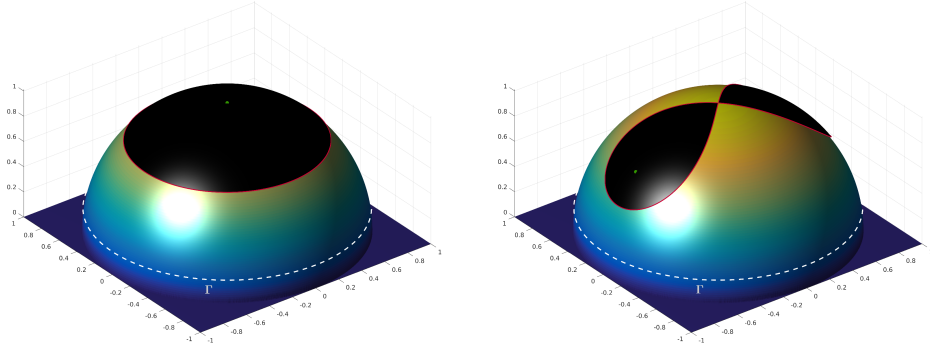
pression des non maxima 3D et une série d’interpolations quadratiques pour extraire des points clés; les deux détecteurs visent à fournir rapidement des invariances aux rotations et à l’échelle. Certains détecteurs de région basés sur les moments [LG94, Bau00], comme les détecteurs de région Harris-Affine et Hessian-Affine [MS02, MS04], un détecteur de régions basé sur les bords [TVO99, TV04], un détecteur de région basé sur l’intensité [TV00, TV04], un détecteur de région basé sur l’entropie [KZB04], le détecteur de régions basé sur deux lignes de niveaux MSER (“maximally stable extremal region”) [MCUP04] et le LLD (“level line descriptor”) [MSCG03, MSC<sup>+</sup>06, CLM<sup>+</sup>08], sont conçus pour être invariants aux transformations affines. Le détecteur MSER, en particulier, a démontré qu’il était souvent plus performant que d’autres détecteurs invariants affines, suivi par les détecteurs Hessian-Affine et Harris-Affine [MTS<sup>+</sup>05].

Dans son article de référence [Low04], Lowe a proposé une transformée de l’image en un ensemble de descripteurs (appelée SIFT), qui est invariante aux changements d’échelle, translations et rotations, et partiellement invariante à l’éclairage et aux changements de point de vue. La méthode SIFT combine un détecteur de points clés DoG qui est invariant à la rotation, la translation et l’échelle (une preuve mathématique de son invariance à l’échelle est donnée dans [MY08]) avec un descripteur basé sur la distribution de l’orientation du gradient dans le voisinage du point clé, qui est partiellement invariant par changements d’illumination et de point de vue [Low04]. Ces deux étapes de la méthode SIFT seront appelées respectivement *détecteur SIFT* et *descripteur SIFT*. Le détecteur SIFT est *a priori* moins invariant aux transformations affines que les détecteurs Hessian-Affine et Harris-Affine [MS01, MS04].

Il a néanmoins été montré que le descripteur SIFT est supérieur à de nombreux autres descripteurs [MS05] tels que les *distribution-based shape context* [ATRB95], les *geometric histogram descriptors* [ATRB95], les *derivative-based complex filters* [Bau00, SZ02], et les invariants basés sur des moments [VMU96]. Un certain nombre de variantes et d’extensions du descripteur SIFT, y compris PCA-SIFT [KS04], GLOH [MS05] et SURF [BTV06] ont été développées depuis [FS07, LÁJA06]. Plus récemment, RootSIFT a été proposé dans [AZ12], et suggère une légère modification des descripteurs SIFT afin de les comparer par un noyau de Hellinger. La plupart de ces variantes de SIFT revendiquent plus de robustesse et de pouvoir de discrimination avec une complexité réduite.

Les descripteurs binaires constituent une autre branche des descripteurs locaux mettant l’accent sur la rapidité et l’efficacité. Dans BRIEF [CLSF10] un patch local flouté entourant le point clé est échantillonné sur un ensemble de paires de pixels aléatoires dont les valeurs sont comparées, produisant ainsi une chaîne binaire. Cela permet d’exploiter les caractéristiques du patch d’une manière différente à celles basées sur des histogrammes comme SIFT, et conduit à une signature compacte, représentée par une séquence de bits. De même, le descripteur BRISK est une chaîne binaire résultant de la différence de luminosité calculée autour du point clé. Une autre alternative de descripteur binaire local est le descripteur LATCH [LH16]. Ses auteurs affirment que son temps d’extraction n’est que légèrement supérieur à d’autres descripteurs binaires et bien plus rapide que les représentations flottantes (“floats”).

Des espaces-échelle non linéaires ont également été utilisés afin de rendre le flou adaptatif à l’image, de manière locale ; ils floutent les petits détails mais préservent les bords des objets. La méthode de description KAZE, introduite dans [ABD12], est censée augmenter la répétabilité et le pouvoir de distinction de SIFT et de SURF grâce à l’utilisation d’un filtre de diffusion non linéaire. Le principal inconvénient de KAZE est la vitesse. Une version accélérée, appelée AKAZE [ANB13], génère des caractéristiques moins exigeantes sur le plan du calcul tout en restant basé sur une diffusion non linéaire.



(a) 45° de visibilité autour d'une vue frontale (b) 45° de visibilité autour d'une vue oblique de 45°

Figure 9: (Points de vue en perspective)

Point vert - position de la caméra requête  
 Ligne pointillée -  $\partial\mathcal{B}([Id], \log 4\sqrt{2})$   
 Surface noire - Positions visibles des caméras de cibles

Les méthodes mentionnées précédemment ont connu beaucoup de succès. Cependant, aucune d'entre elles n'est totalement invariante affine. Comme indiqué dans [Low04], les approches Harris-Affine et Hessian-Affine débutent par des échelles de caractéristiques initiales et des positions sélectionnées d'une manière qui n'est pas elle-même invariante affine. Le fait que le flou optique et les transformées affines ne commutent pas, comme signalé dans [YM11], explique la performance limitée de l'invariance affine de la normalisation des méthodes MSER, LLD, Harris-Affine et Hessian-Affine. Comme indiqué dans [CLM<sup>+</sup>08], MSER et le LLD ne sont pas invariants à l'échelle : ils ne s'adaptent pas aux changements drastiques de la structure des lignes de niveau causés par le flou. SIFT (et ses variantes directes) est en fait la seule méthode qui soit totalement invariante à l'échelle. Cependant, puisque SIFT n'est pas conçu pour couvrir l'ensemble de l'espace affine, ses performances chutent rapidement en cas de changement de point de vue important. En fait, tous les descripteurs cités ne sont pas totalement invariants aux changements de point de vue ; ils continuent à fonctionner pour des angles de vue allant jusqu'à 60° pour les objets plats [YM11, MMP15], mais leur performance chute drastiquement pour les angles supérieurs à 45° [Kar16]. Dans la Figure 9, nous montrons des positions fixes de la caméra de requête (points verts) et toutes les positions possibles de la caméra cible (surfaces noires) avec des variations d'angle de vue inférieures à 45°.

Dans cette thèse, la question de l'invariance au point de vue pour la mise en correspondance d'images est centrale. Comme beaucoup d'autres dans la littérature, nous atteignons cet objectif par invariance affine locale, invariance qui peut être obtenue en exploitant les propriétés affines à différents endroits de la démarche de mise en correspondance. Nos principales contributions concernant la mise en correspondance d'images en cas de fortes variations de point de vue sont présentées dans le Tableau 5. Nous énumérons dans le Tableau 6 plusieurs outils et méthodes conçus pour valider nos affirmations. Le Tableau 7 présente toutes les techniques apparaissant dans cette thèse pour atteindre une invariance affine, et en définitive, une invariance au point de vue. Enfin, le Tableau 8 énumère toutes les méthodes de mise en correspondance par invariance affine apparaissant dans cette thèse.

**Table 5** Principales contributions à la comparaison d’images dans le cadre de forts changements de points de vue.

Propositions	Chap. 1	Chap. 2	Chap. 3	Chap. 5	Chap. 6	Chap. 7
Formules analytiques pour mesurer les déformations induites par changement de point de vue	✓					
Recouvrements optimaux dans l’espace de tilts	✓	✓				✓
Appariements distinctifs capables de saisir des structures répétitives avec changement de point de vue		✓	✓	✓		
Descripteurs invariants aux points de vue au-delà de 60°				✓		
Estimations de transformations de la géométrie locale					✓	
Réduction de la complexité	✓	✓		✓		✓

**Table 6** Outils et méthodes pour la validation des propriétés affines.

Validation	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
Données synthétisées	✓		✓		✓	✓		✓
Récupération réussie d’homographies sur des images réelles	✓	✓	✓	✓	✓	✓	✓	✓
Récupération réussie d’homographies dans plus d’une base de données				✓		✓	✓	✓
Estimations de la densité sur les mesures des descripteurs					✓			✓
Estimation de la densité des paramètres affines						✓		
Estimations de la densité dans l’espace des tilts							✓	
Courbes ROC								✓

**Table 7** Techniques pour atteindre l'invariance affine.

Techniques	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
Simulations optiques affines	✓	✓	✓	✓	✓		✓	✓
Descriptions directes					✓	✓	✓	✓
Normalisation des patches						✓	✓	✓

**Table 8** Méthodes disponibles pour la comparaison d'images invariante affine. S'il n'y a aucune référence indiquée, cela signifie que la méthode a été proposée dans cette thèse. Entre accolades, tous les descripteurs possibles pour un réglage fixe.

Méthodes	Chap. 1	Chap. 2	Chap. 3	Chap. 4	Chap. 5	Chap. 6	Chap. 7	Chap. 8
ASIFT [MY09]	✓			✓			✓	
FAIR-SURF [PLYP12]	✓	✓						
Optimal Affine-RootSIFT	✓	✓	✓	✓	✓		✓	
Optimal Affine-RootSIFT Revisited		✓		✓				
Optimal Affine-SURF	✓	✓		✓				
Optimal Affine-{SIFT, BRISK, BRIEF, ORB, AKAZE, LATCH, FREAK, AGAST, LUCID, DAISY}	✓	✓						
Optimal Affine-{LDA64, LDA128, DIF64, DIF128, HalfRootSIFT, HalfSIFT}		✓						
Optimal Affine-{AC, AC-W, AC-Q}			✓	✓				
SIFT-AID					✓	✓	✓	
SIFT+Affnet+HardNet						✓		
HesAffNet [MRM18]							✓	✓
Adaptive-ARootSIFT, Greedy-ARootSIFT							✓	
HessAff-{AID, AID21}								✓
HessAffnet-{AID, AID21}								✓

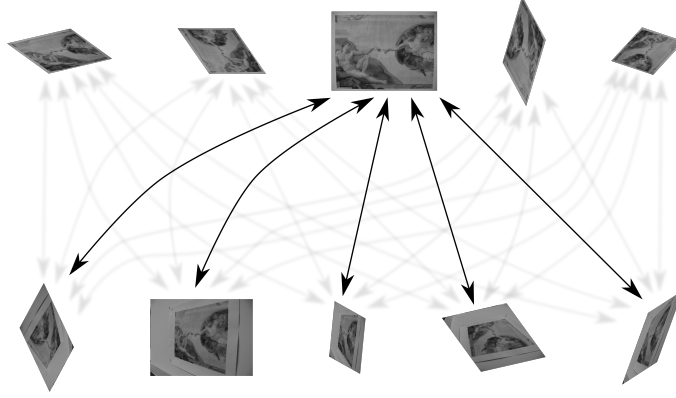


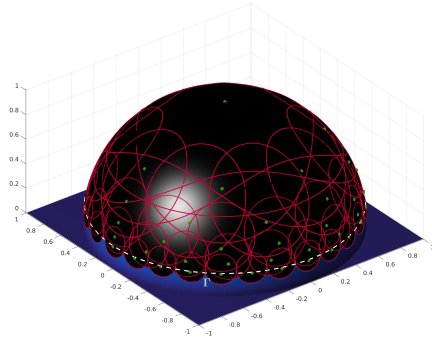
Figure 10: Les algorithmes IMAS commencent par appliquer un ensemble fini de simulations affines optiques à  $u$  et  $v$ , suivies de comparaisons par paires.

## Recouvrement de l'espace des tilts

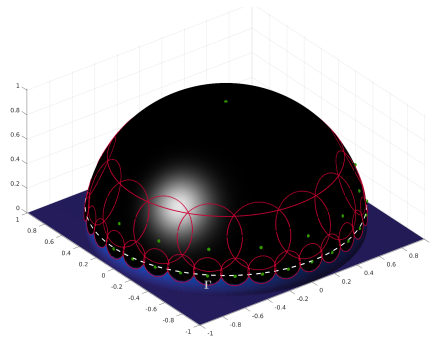
Dans le Chapitre 1, nous commençons par calculer des formules exactes pour déterminer les surfaces d'accessibilité noires affichées dans la Figure 9. Cela revient à obtenir une formule pour la distance qui mesure la déformation par tilt. Nous proposons ensuite une méthode mathématique pour analyser les nombreux algorithmes mettant en correspondance les images par simulation affine (IMAS). Pour devenir invariant affine, ces algorithmes appliquent un ensemble discret de transformations affines aux images, avant de comparer toutes les images obtenues par une comparaison d'images invariante à l'échelle (SIIM) comme SIFT (voir la Figure 10). De toute évidence, cette multiplication d'images à comparer augmente la complexité de la mise en correspondance. Trois questions se posent alors : a) quel est le meilleur ensemble de transformations affines à appliquer à chaque image pour obtenir une invariance affine pratique et complète ? b) quelle est la complexité la plus faible possible pour la méthode résultante ? c) comment choisir la méthode SIIM sous-jacente ? Nous fournissons une réponse explicite et une preuve mathématique de la quasi-optimalité de la solution à la première question. La Figure 11 souligne les avantages de cette méthode en optimisant l'ensemble des simulations affines de la méthode classique ASIFT [MY09, YM11]. En réponse à la question b), nous constatons que le rapport de complexité quasi optimal entre l'appariement affine complet et l'appariement invariant à l'échelle est réduit de plus de moitié par rapport aux méthodes IMAS actuelles. Cela signifie que le nombre de points clés nécessaires à l'appariement affine peut être réduit de moitié et que la complexité de l'appariement est divisée par quatre pour obtenir exactement les mêmes performances. Cela signifie également qu'un ensemble de descripteurs invariants affines peut être associé à n'importe quelle image. Le prix à payer pour une invariance affine complète est que le cardinal de cet ensemble est environ 6,4 fois plus grand que pour un SIIM.

## Mise en correspondance rapide et invariante affine d'images

Le Chapitre 2 se concentre principalement sur les détails de mise en oeuvre des méthodes proposées au chapitre précédent, avec deux modifications structurelles et quelques autres améliorations des méthodes IMAS. Comme indiqué précédemment, ces méthodes



(a) ASIFT [MY09, YM11], un recouvrement de  $56^\circ$  avec un ensemble de 41 simulations affines qui rendent les points de vue des caméras cibles visibles jusqu'à  $80^\circ$ . Hautement redondant, il recouvre néanmoins la région pour laquelle il a été conçu.



(b) Notre proposition, un recouvrement de  $56^\circ$  avec un ensemble de 28 simulations affines qui rendent les points de vue des caméras cibles visibles jusqu'à  $82^\circ$ .

Figure 11: Recouvrements

Points verts - Simulations de caméra affine  
Lignes rouges - Tolérance de visibilité de chaque simulation affine  
Surfaces noires - Régions de points de vue visibles  
Ligne pointillée - Régions recouvertes

atteignent l'invariance affine en appliquant un ensemble fini de transformations affines aux images avant de les comparer avec une méthode SIIM comme SIFT ou SURF. Nous décrivons dans le Chapitre 2 comment optimiser ces méthodes IMAS. Tout d'abord, nous détaillons un algorithme calculant un ensemble discret minimal de transformées affines à appliquer à chaque image avant la comparaison. Cela permet d'obtenir une invariance affine pratique, complète, au coût de calcul le plus bas possible. La complexité de l'appariement des algorithmes IMAS actuels est divisée par 4 environ. Notre approche associe également à chaque image un ensemble de descripteurs invariants affines, qui est deux fois plus petit que l'ensemble de descripteurs habituellement utilisés dans les méthodes IMAS, et seulement 6,4 fois plus grand que l'ensemble de descripteurs invariants affines de méthodes SIIM. Afin de réduire le nombre de fausses correspondances, qui sont intrinsèquement plus fréquentes dans les méthodes IMAS que dans les méthodes SIIM, nous introduisons la notion d'hyper-descripteur, qui regroupe les descripteurs dont les points clés sont spatialement proches. Les hyper-descripteurs visent à fournir différentes représentations affines d'une scène commune. La Figure 12 montre trois zones affines qui doivent être décrites et regroupées dans un hyper-descripteur. Enfin, nous proposons également un critère de comparaison permettant de faire correspondre chaque point clé de l'image requête avec plusieurs points clés de l'image cible, afin de traiter les situations où un objet est répété plusieurs fois dans l'image cible. Une démo en ligne permettant de reproduire tous les résultats est disponible dans l'article d'IPOL

<https://ipolcore.ipol.im/demo/clientApp/demo.html?id=225>,

où le code source est également mis à disposition. Les instructions de compilation et d'utilisation sont incluses dans le fichier README.md de l'archive. Des informations complémentaires sur le Chapitre 2 sont disponibles sur la page web :

<https://rdguez-mariano.github.io/pages/hyperdescriptors>.

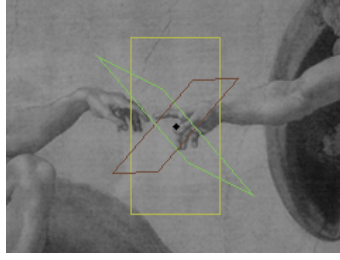


Figure 12: Un hyper-descripteur  $d = \{\omega_1, \omega_2, \omega_3\}$

Point noir - Centre de  $d$  et de chaque  $\omega_i$   
 Parallélogrammes colorés - Vues affines décrites par  $\omega_i$   
 (Les vues affines sont des patchs carrés des régions désignées par les parallélogrammes)

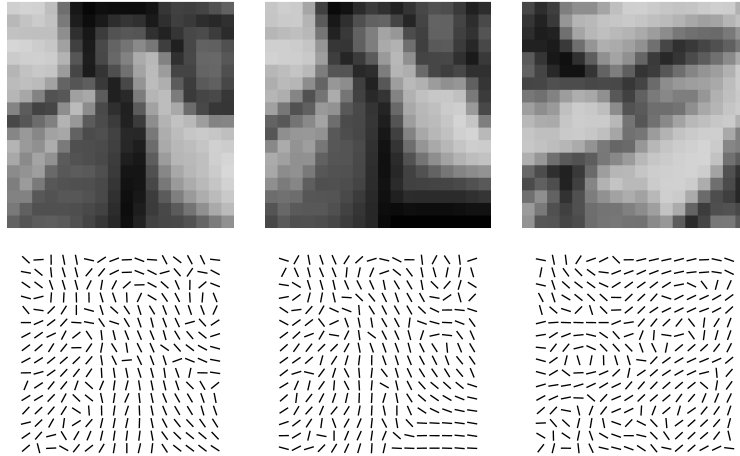


Figure 13: Trois patches d'image et leurs champs d'orientation correspondants utilisés comme descripteurs. Les deux premiers sont similaires, tandis que le troisième est différent.

## Comparaison d'images avec structures répétitives

Dans le Chapitre 3, nous nous concentrons sur le problème de la comparaison invariante affine d'images en présence de bruit et de structures répétitives. On conserve le schéma classique des points clés, descripteurs et appariement. Un champ local d'orientations du gradient d'image est utilisé comme descripteur (voir Figure 13) et deux méthodes d'appariement sont proposées, sur la base d'une approche *a contrario*, pour le traitement des structures répétitives. L'invariance affine est obtenue par des simulations. Les méthodes proposées permettent d'obtenir des performances compétitives dans le cadre de structures répétitives. Des informations complémentaires pour le Chapitre 3 sont disponibles sur la page web :

<https://rdguez-mariano.github.io/pages/acdesc>.

## AID : Un descripteur invariant affine

Le Chapitre 5 s'intéresse toujours à la comparaison invariante affine, mais cette fois-ci à l'aide de réseaux de neurones. Un descripteur encode les informations locales autour d'un point clé. Un avantage de ces approches locales est que les déformations des points

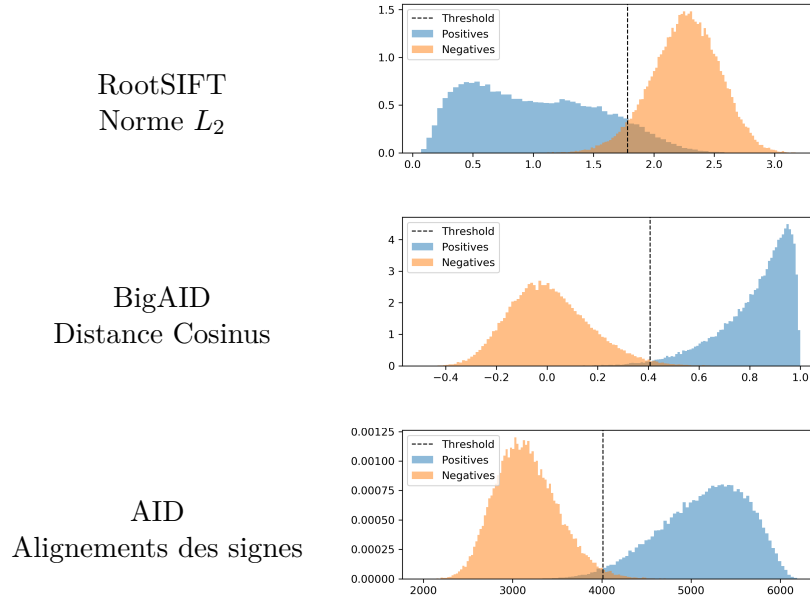


Figure 14: Estimation de la densité pour les mesures dans les cas positifs et négatifs. Pour cela,  $6 \cdot 10^5$  paires de patches aléatoires intra-classe et extra-classe ont été utilisées. La ligne verticale représente le seuil minimisant les deux probabilités d’erreur : faux négatifs et faux positifs.

de vue sont bien approximées par des transformations affines locales. Cela a motivé la quête sans fin de descripteurs locaux invariants affines. Malgré de nombreux efforts, de tels descripteurs sont restés inatteignables, ce qui a finalement poussé à un compromis entre l’utilisation de simulations de points de vue et la normalisation de patches pour atteindre une vraie invariance affine. Dans le Chapitre 5, nous proposons un descripteur de patch basé sur un réseau convolutif qui capture l’invariance affine sans avoir besoin de simulations de point de vue ou de normalisation de patch. Ceci est rendu possible en entraînant un réseau de neurones à associer des représentations vectorielles similaires à des patches liés par des transformations affines. Lors de l’appariement, ces vecteurs sont comparés très efficacement. L’invariance de la méthode à la translation, la rotation et l’échelle est encore obtenue par les premières étapes de SIFT, qui produisent les points clés. Le descripteur proposé surpasse les meilleurs descripteurs pour la conservation des propriétés d’invariance affine. La Figure 14 montre les estimations de densité pour les paires de patches positifs et négatifs pour RootSIFT, et nos descripteurs BigAID et AID. Des informations complémentaires sur le Chapitre 5 sont disponibles sur la page web :

<https://rdguez-mariano.github.io/pages/siftaid>.

## Estimation robuste des transformations affines locales

Les coordonnées des points correspondants déterminées par les approches classiques de comparaison d’images définissent des approximations locales d’ordre zéro de la transformation globale entre deux images. Mais les patches autour des points clés contiennent généralement plus d’informations, qui peuvent être exploitées pour obtenir une approximation du premier ordre de la transformation, en incorporant des transformations affines locales entre les points clés correspondants (voir la Figure 15 pour une représentation visuelle de ces approximations du premier ordre). Dans le Chapitre 6, nous proposons



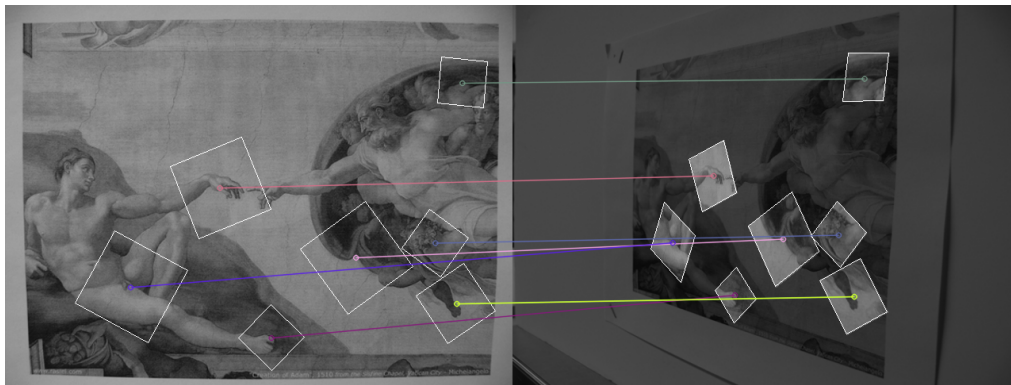


Figure 15: Quelques correspondances ainsi que des transformations affines locales estimées par le réseau LOCATE proposé. Les patches sur la cible sont des versions transformées de leur patch de requête correspondant.

une méthode d’estimation de la transformation affine locale (LOCATE) apprise par un réseau de neurones. Nous montrons que LOCATE améliore considérablement la précision de l’estimation de la géométrie locale en retrouvant aussi les transformations inverses. Une deuxième contribution sur la mise en correspondance guidée et le raffinement est également présentée. La nouveauté consiste ici à utiliser LOCATE pour proposer des nouvelles correspondances SIFT avec des localisations, orientations et échelles précises. Nos expériences montrent que le gain en précision fourni par LOCATE joue un rôle important dans des applications telles que la mise en correspondance guidée. La troisième contribution de ce chapitre consiste en une modification de l’algorithme RANSAC, qui utilise LOCATE pour améliorer l’estimation de l’homographie entre une paire d’images. Ces approches sont plus performantes que RANSAC pour différents choix de descripteurs et de bases de données d’images, et permettent d’augmenter la probabilité de succès dans l’identification de paires d’images dans des bases de données exigeantes. Des informations complémentaires sur le Chapitre 3 sont disponibles sur la page web :

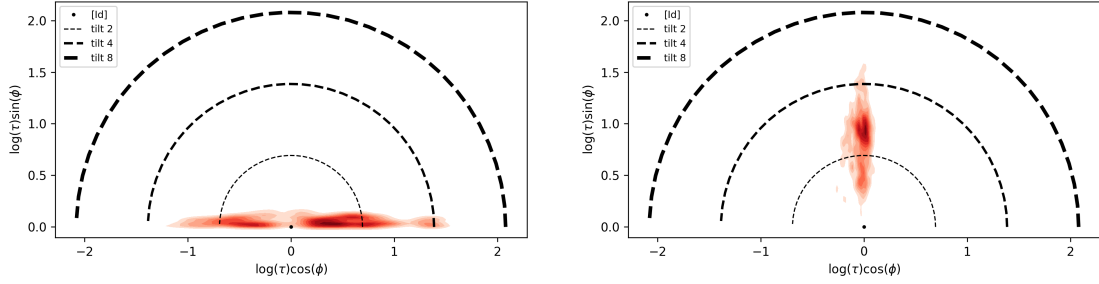
<https://rdguez-mariano.github.io/pages/locate>.

## Recouvrements assistés par CNN dans l’espace des tilts

Comme indiqué ci-dessus, les descripteurs invariants affines restent inatteignables, ce qui explique le succès et le développement des méthodes IMAS. Ces méthodes simulent les changements de point de vue pour atteindre l’invariance souhaitée. Pourtant, les méthodes récentes basées sur CNN semblent fournir un moyen d’apprendre des descripteurs invariants affines. Dans le Chapitre 3, en guise de première contribution, nous montrons que les méthodes actuelles basées sur des CNN sont loin d’atteindre les performances des approches IMAS. Cela confirme que les méthodes apprises peuvent encore être améliorées. Deuxièmement, nous montrons que les récentes avancées en matière de normalisation des patches affines par CNN peuvent être utilisées pour créer des méthodes IMAS adaptatives qui sélectionnent leurs simulations affines en fonction des images requête et cible. La Figure 16 montre les estimations de la densité dans l’espace des tilts (formellement introduit dans le Chapitre 1) pour les images requête et cible dans la paire ‘cat’ de la base de données EVD [MMP15]. On remarque la concentration autour de directions orthogonales dans l’espace des tilts des transformations affines fournies par Affnet [MRM18] à partir des images de requête et de cible. Rien qu’en regardant ces densités, on peut déjà déduire



(a) Objet commun aux images requête (à gauche) et cible (à droite).



(b) Estimations de la densité par noyau des transformations affines d’Affnet [MRM18] de la requête (à gauche) et de la cible (à droite).

Figure 16: Estimations de la densité par noyaux dans l’espace des tilts des cartes affines extraites par Affnet [MRM18] pour la paire d’images ‘cat’ de la base de données EVD [MMP15].

que l’objet commun aux deux images a été vu à partir de positions de caméra qui diffèrent de  $90^\circ$ . En pratique, les prédictions d’Affnet [MRM18] seront utilisées pour sélectionner des transformations affines convenables à tester dans les méthodes IMAS. Les méthodes hybrides ainsi proposées se révèlent être un bon compromis : d’une part, elles atteignent les performances des meilleures méthodes IMAS mais sont plus rapides ; d’autre part, elles sont nettement plus performantes que les méthodes sans simulation, y compris les plus récentes. Des informations complémentaires sur le Chapitre 3 sont disponibles sur la page web :

<https://rdguez-mariano.github.io/pages/adimas>.

## Articles publiés

Les documents publiés correspondant aux travaux de cette thèse sont énumérés ci-dessous.

1. [RDM18a]. M. Rodriguez, J. Delon, and J.-M. Morel. Covering the space of tilts: application to affine invariant image comparison. In *SIIMS*, 11(2):1230–1267, 2018.
2. [RDM18b]. M. Rodriguez, J. Delon, and J.-M. Morel. Fast affine invariant image matching. In *IPOLE*, 8:251–281, 2018.
3. [RGvG18]. M. Rodriguez and R. Grompone von Gioi. Affine invariant image comparison under repetitive structures. In *ICIP*, pages 1203–1207, Oct 2018.
4. [RFGvG<sup>+</sup>19]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, J.-M. Morel, and J. Delon. SIFT-AID: boosting SIFT with an affine invariant descriptor based on convolutional neural networks. In *ICIP*, Sep 2019.

5. [RFGvG<sup>+</sup>20]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, and J. Delon. Robust estimation of local affine maps and its applications to image matching. In *WACV*, 2020.
6. [RFvG<sup>+</sup>20]. M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, J. Delon, and J.-M. Morel. CNN-assisted coverings in the space of tilts: best affine invariant performances with the speed of CNNs. In *ICIP*, Oct 2020.

Les codes sources ont été publiés sur github.

1. Le cadre de test, les figures et l’optimiseur de recouvrements apparaissant dans [RDM18a] peuvent être trouvés à l’adresse:

[https://github.com/rdguez-mariano/imas\\_analytics](https://github.com/rdguez-mariano/imas_analytics)

2. Toutes les méthodes IMAS présentées dans [RDM18b, RGvG18] peuvent être trouvées à l’adresse suivante:

[https://github.com/rdguez-mariano/fast\\_imas\\_IPOL](https://github.com/rdguez-mariano/fast_imas_IPOL)

3. La méthode SIFT-AID ainsi que tous les résultats apparaissant dans [RFGvG<sup>+</sup>19] peuvent être trouvés à l’adresse suivante:

<https://github.com/rdguez-mariano/sift-aid>

4. La méthode LOCATE, plus deux applications (appariement affine guidé et une version de RANSAC homographique), quelques autres méthodes de correspondance (par exemple HessAff+Affnet+HardNet, HessAff+AID et SIFT+Affnet+HardNet) et tous les résultats apparaissant dans [RFGvG<sup>+</sup>20] peuvent être trouvés à l’adresse suivante:

<https://github.com/rdguez-mariano/locate>

5. Les méthodes IMAS adaptatives figurant dans [RFvG<sup>+</sup>20] peuvent être trouvées à l’adresse suivante :

[https://github.com/rdguez-mariano/fast\\_imas\\_IPOL/tree/master/adaptiveIMAS](https://github.com/rdguez-mariano/fast_imas_IPOL/tree/master/adaptiveIMAS)

## Part I

# Image Matching by Affine Simulations



# 1 Covering the space of tilts

## 1.1 Introduction

Image matching, which consists in detecting shapes common to two images, is a crucial preliminary step of a large number of computer vision applications, such as scene recognition [VGS10, BS11, SAS07] and detection [FSKP, NTG<sup>+</sup>06], object tracking [ZYS09], robot localization [SLL01, VL10, MMK06], image stitching [AAC<sup>+</sup>06, BL03], image registration [YSST07, LYT11] and retrieval [HL04, GLGP13], 3D modeling and reconstruction [Fau93, GZS11, VV05, AFS<sup>+</sup>11], motion estimation [WRHS13], photo management [SSS06], symmetry detection [LE06] or even image forgeries detection [CPV15]. The problem has implementation variants depending on the set up. If for example the user knows that both compared images are related, the focus is on detecting the most reliable common set of shape descriptors. In the detection set up, an image is compared to a database of images and the question is to retrieve related images in the database. This is for example crucial for performing video search [SSR<sup>+</sup>09]. Local shape descriptors must be extracted for this purpose, and this description should be as invariant as possible to viewpoint changes and of course as sparse as possible. In our discussion we will most of the time refer to the simpler set up where two images are being compared. But the reduction of the number of descriptors is of course still more important for comparing an image to an image database as initially proposed in [SZ<sup>+</sup>03]. In this last reference, large sets of descriptors are sparsified by clustering techniques. This only indicates how important it is to reduce as much as possible the set of affine descriptors of each image.

**Detectors, descriptors and affine invariance** Given a query image of some physical object and a set of target images, the first goal of image matching is to decide if these target images contain a view of the same object. If the answer is positive, image matching aims at localizing this object in these target images. Deciding if the object is present is difficult and becomes especially tricky for large image databases, for which the control of false matches is crucial. Another difficulty of the matching problem comes from the change of camera viewpoints between images. In order to cope with these viewpoint changes, the whole matching process should be as invariant as possible to the resulting image deformations. As we shall develop, this requires affine invariance for the recognition process.

The classical approach to image matching consists in three steps: detection, description and matching. First, keypoints are detected in the compared images. Second, regions around these points are described and encoded in local invariant descriptors. Finally, all these descriptors are compared and possibly matched. Using local descriptors yields robustness to context changes. Both the detection and description steps are usually designed

to ensure some invariance to various geometrical or radiometric changes.

Local image point detectors are always translation invariant. While the venerable Harris point detector [HS88] is only invariant to translations and rotations, the Harris-Laplace [MS01], Hessian-Laplace [MS04] or DoG (Difference-of-Gaussian) region detectors [Low04] are invariant to similarity transformations, *i.e.* translations, rotations and scale changes. To ensure invariance to affine transforms, some authors have proposed moment-based region detectors [LG94, Bau00] including the Harris-Affine and Hessian-Affine region detectors [MS02, MS04]. Locally affine invariant region detectors can also be based on edges [TV09, TV04], intensity [TV00, TV04], or entropy [KZB04]. Finally, the detectors MSER (“maximally stable extremal region”) [MCUP04] and LLD (“level line descriptor”) [MSCG03, MSC<sup>+</sup>06, CLM<sup>+</sup>08] both rely on level lines. Yet the affine invariance of these detectors is limited by the fact that optical blur and affine transforms do not commute, as pointed out in [MY09]. Level line based detectors like MSER therefore are not fit to handle scale changes. Indeed, they do not take into account the effect of blur on the level line geometry [CLM<sup>+</sup>08].

In the last 15 years, numerous invariant image descriptors have been proposed in the literature, but the most well-known and the most widely used remains the scale-invariant feature transform (SIFT), introduced by Lowe in his landmark paper [Low04]. SIFT makes use of a DoG region detector. It is fully invariant to similarities (see [MY08] for a mathematical proof of this fact). Each *SIFT descriptor* is composed of histograms of gradient orientation around a key point, invariant to local radiometric changes and to geometrical image similarities. As a result, the SIFT method can be considered as partially invariant to illumination, fully invariant to geometrical similarities. But its success is certainly also due to its robustness to reasonable viewpoint changes.

The superiority of SIFT based descriptors has been demonstrated in several comparative studies [MS05, MP07]. As a consequence, many variants of the SIFT descriptor have emerged, among which we can mention PCA-SIFT [KS04], GLOH (gradient location-orientation histogram) [MS05], SURF (speeded up robust features) [BTV06] or RootSIFT [AZ12]. The main claims of these variants are a lower complexity or a greater robustness to viewpoint changes. In the same vein, binary descriptors have also received much attention. Focusing on speed and efficiency, the BRIEF [CLSF10], BRISK [LCS11] or LATCH [LH16] descriptors are compact and represented by sequences of bits, and can be compared more quickly than floating point descriptors like those used in SIFT. Descriptors based on nonlinear scale spaces, such as KAZE [ABD12] or its accelerated version AKAZE [ANB13], have also been proposed to locally adapt blur to the image data.

None of the previously mentioned state-of-the-art methods is fully affine invariant. The SIFT method does not cover the whole affine space and its performance drops under substantial viewpoint changes. SIFT and the other aforementioned descriptors cannot cope with viewpoint differences larger than 60° for planar objects [MY09, MMP15], and are still usable but much less efficient for angles larger than 45° [Kar16]. We shall give and use here concrete measurements of their resilience to view angle changes.

To overcome this limitation, several simulation-based solutions have been recently proposed. The core idea of these algorithms, that we choose to call by the generic term **IMAS** (Image Matching by Affine Simulation), is to simulate a set of views from the initial images, by varying the camera orientation parameters. These simulations allow to capture far stronger viewpoint angles than standard matching approaches, up to 88°. Among those IMAS algorithms, we can mention ASIFT [YM11], FAIR-SURF [PLYP12] and MODS [MMP15].

A first suggestion to simulate affine distortions before applying a **SIIM** (Scale Invariant Image Matching) appeared in [PH03] where the authors proposed to simulate two tilts and two shear deformations followed by SIFT in a cloth motion capture application. As argued in [YM11, MMP15, PLYP12], if a physical object has a smooth or piecewise smooth boundary, its views obtained by cameras in different positions undergo smooth apparent deformations. These regular deformations are locally well approximated by affine transforms of the image plane. By focusing on local image descriptors, the changes of aspect of objects can therefore be modeled by affine image deformations.

The problem of constructing affine invariant image descriptors by using an affine Gaussian scale space, that is equivalent to simulating affine distortions followed by the heat equation, has a long story starting with [Iij71, Blo92, Lin93, LG94]. The idea of affine shape adaptation underlying one of the methodologies for achieving affine invariance, was then in turn used as a base for the work on affine invariant interest points and affine invariant matching in [LG94, Bau00, MS02, MS04, TVO99, TV04, TV00]. The notion of an affine invariant reference frame was further developed in [Lin11, Lin13a]. Nevertheless, to the best of our knowledge, the direct constructions of affine invariant descriptors as fixed points for an iterative affine normalization process have never found a mathematical justification.

The first IMAS method provided with a mathematical proof of affine invariance is ASIFT [MY09, YM11]. The authors of this paper proposed it as an affine invariant extension of SIFT and proved it to be fully affine invariant in a continuous model. The structure of ASIFT is generic in the sense that it can be implemented with any local descriptor, provided this descriptor has some robustness to viewpoint changes like the SIFT descriptors. Unlike MSER, LLD, Harris-Affine and Hessian-Affine, which attempt at normalizing all of the six affine parameters, ASIFT simulates three parameters and normalizes the rest. More specifically, ASIFT simulates the two camera axis parameters, and then applies SIFT which simulates the scale and normalizes the rotation and the translation. Of the six parameters required for affine invariance, three are therefore simulated and three normalized.

Two recent successful methods follow the same affine simulation path. FAIR-SURF [PLYP12] combines the affine invariance of ASIFT and the efficiency of SURF. The MODS image comparison algorithm introduced in [MMP15] also relies on this principle and affine simulations are generated on-demand if needed in the process of comparing two images. MODS employs a combination of different detectors when comparing images. It outperforms state-of-the-art image comparison approaches both in affine robustness and speed.

Other IMAS approaches without local descriptors have also been put up for template matching. FAsT-Match [KRTA13] delivers affine invariance by assuming that the template (a patch in the query image) can be recovered inside the target image by a *unique* affine map. Meaning there is no subjacent projective map to identify. Contrary to IMAS with local descriptors, the six required parameters to attain affine invariance are simulated instead of the three used in the present chapter.

In this chapter, we are interested in generic IMAS algorithms based on local descriptors and in their geometric optimization. In order to measure the degree of viewpoint change between different views of the same scene, we draw on the concept of *absolute and relative transition tilts*, previously introduced in [MY09, YM11], and we illustrate why simulating large tilts on both compared images is necessary to obtain a fully affine invariant recognition. Indeed, transition tilts can in practice be much larger than absolute tilts, since they may behave like the square of absolute tilts.



The key question of IMAS methods is how to choose the list of affine transforms applied to the images before comparison. This list should be as short as possible to limit the computing time. But it should also sample the widest possible range of affine transforms. As we shall see, this question is closely related to the question of finding optimal coverings of the space of affine tilts. This question is formalized and solved in Section 1.2, where we find nearly optimal coverings. Section 1.3 applies this result to IMAS algorithms. It first presents a complete mathematical theory of IMAS algorithms, proving that they are fully affine invariant under the assumption that the underlying SIIM has a (quantifiable) limited affine invariance. Section 1.4 gives an experimental validation. It starts by measuring the exact extent of affine invariance for several SIIMs and deduces the corresponding complexity required to attain full affine invariance from each.

## 1.2 The space of affine tilts

In this section, we introduce the space of tilts for planar affine transforms, and we look for optimal coverings of this space. Optimal coverings will be used in the next section to define an optimal discrete set of affine transformations as the basis for IMAS algorithms. The rest of this section can be read as a sequence of purely geometric results. However, the reader might prefer to keep in mind that the affine transforms considered here can be interpreted as different viewpoints of a camera, or more generally as the transition from an image taken from a viewpoint to an image taken from another viewpoint. Indeed, given a frontal snapshot of a planar object  $u(\mathbf{x}) = u(x, y)$ , we can transition from any affine view  $Bu$  of the same object to any other affine view  $Au$  through the affine transformation  $AB^{-1}$ . This requires some notation. For any linear invertible map  $A \in GL^+(2)$ , we denote the affine transform  $A$  of a continuous image  $u(\mathbf{x})$  by  $Au(\mathbf{x}) = u(A\mathbf{x})$ . We recall classic notation for three subsets of the general linear group  $GL(2)$  of invertible linear maps of the plane,

$$\begin{aligned} GL^+(2) &= \{A \in GL(2) \mid \det(A) > 0\}, \\ GO^+(2) &= \{A \in GL^+(2) \mid A \text{ is a similarity}\}, \\ GL_*^+(2) &= GL^+(2) \setminus GO^+(2), \end{aligned}$$

where we call similarity any combination of a rotation and a zoom, and the symbol  $\setminus$  denotes the set difference operator. Our central notion in the discussion is the *tilt* of an affine transform, which we now define.

### 1.2.1 Absolute tilts

**Proposition 1.1** ([MY09]). *Every  $A \in GL_*^+(2)$  is uniquely decomposed as*

$$A = \lambda R_1(\psi) T_t R_2(\phi) \tag{1.1}$$

where  $R_1, R_2$  are rotations and  $T_t = \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix}$  with  $t > 1$ ,  $\lambda > 0$ ,  $\phi \in [0, \pi[$  and  $\psi \in [0, 2\pi[$ .

**Remark 1.1.** A similar decomposition to (1.1) was also presented in [Lin95] for small deformations around the identity.

**Remark 1.2.** It follows from this proposition that any affine map  $A \in GL^+(2)$  is either uniquely decomposed as in (1.1) or is directly expressed as a similarity  $\lambda R_1$ .

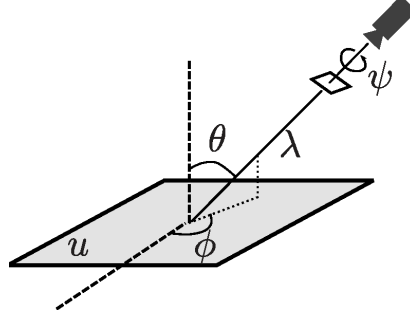


Figure 1.1: Geometric Interpretation of (1.1)

Figure 1.1 shows a camera viewpoint interpretation of this affine decomposition where the longitude  $\phi$  and latitude  $\theta = \arccos \frac{1}{t}$  characterize the camera's viewpoint angles,  $\psi$  parameterizes the camera spin and  $\lambda$  corresponds to the zoom. In the ideal affine model, the camera is supposed to stand at infinite distance from a flat image  $u$ , so that the deformation of  $u$  induced by the camera indeed is an affine map. But the above approximation is still valid provided the image's size is small with respect to the camera distance. In other terms the affine model is locally valid for each small and approximately flat patch of a physical surface photographed by a camera at some distance. Yet, the affine deformation of the object's aspect will be different for each of its patches. This explains why affine invariant recognition methods deal with local descriptors. The parameter  $t$  defined above measures the so-called *absolute tilt* between the frontal view and a slanted view. The uniqueness of the decomposition in (1.1) justifies the next definition.

**Definition 1.1.** We call *absolute tilt* of  $A$  the real number  $\tau(A)$  defined by

$$\begin{cases} GL^+(2) & \rightarrow [1, \infty[ \\ A & \mapsto \begin{cases} 1 & \text{if } A \in GO^+(2) \\ t & \text{if } A \in GL_*^+(2) \end{cases} \end{cases}$$

where  $t$  is the parameter found when applying Proposition 1.1 to  $A$ .

**Proposition 1.2.** Let  $A \in GL^+(2)$ . Then

$$\tau(A) = \sqrt{\frac{\lambda_1}{\lambda_2}} = \|A\|_2 \|A^{-1}\|_2$$

where  $\lambda_1 \geq \lambda_2$  are the singular values of  $A$  and  $\|\cdot\|_2$  is the usual Euclidean matrix norm.

*Proof.* The case of a similarity being straightforward, suppose that  $A \in GL_*^+(2)$ . Then, using (1.1) we can re-write

$$A = R_1 \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} R_2$$

where  $R_1, R_2$  are two rotations and  $\gamma_1 \geq \gamma_2 > 0$ . So

$$A^*A = R_2^t \begin{pmatrix} \gamma_1^2 & 0 \\ 0 & \gamma_2^2 \end{pmatrix} R_2$$

whose eigenvalues are

$$\lambda_1 = \gamma_1^2 \text{ and } \lambda_2 = \gamma_2^2$$

but  $\gamma_1, \gamma_2 > 0$  imply

$$A = \sqrt{\lambda_2} R_1 \begin{pmatrix} \sqrt{\frac{\lambda_1}{\lambda_2}} & 0 \\ 0 & 1 \end{pmatrix} R_2$$

and finally  $\tau(A) = \sqrt{\frac{\lambda_1}{\lambda_2}}$ . In addition, it is well known that

$$\|A\|_2 = \sqrt{\rho(A^*A)} = \sqrt{\lambda_1},$$

$$\|A^{-1}\|_2 = \sqrt{\rho((AA^*)^{-1})} = \frac{1}{\sqrt{\lambda_2}}$$

where  $\rho(A^*A)$  is the largest eigenvalue of  $A^*A$ , i.e, the largest singular value of  $A$ .  $\square$

### 1.2.2 Transition Tilts

Image descriptors like those proposed in the SIFT method are invariant to translations, rotations and Gaussian zooms, which in terms of the camera position interpretation (see Figure 1.1) correspond to a fronto-parallel motion of the camera, a spin of the camera and to an optical zoom. We shall focus on the last part  $T_t R_2$  of the decomposition (1.1) because it is the one that is imperfectly dealt with by SIIMs. SIIMs are instead able to detect objects *up to a similarity*. This leads us to the next definition.

**Definition 1.2.** Let  $A, B \in GL^+(2)$ . Then we define the right equivalence relation  $\sim$  as

$$A \sim B \Leftrightarrow AB^{-1} \in GO^+(2).$$

**Remark 1.3.** It is important to notice here that the right and left equivalence relations do differ. For example, take

$$A = T_2 R_{\frac{\pi}{4}} \text{ and } B^{-1} = R_{\frac{\pi}{4}} T_2,$$

then

$$AB^{-1} = 2R_{\frac{\pi}{2}} \in GO^+$$

whereas

$$B^{-1}A = R_{\frac{\pi}{4}} T_4 R_{\frac{\pi}{4}} \notin GO^+.$$

**Definition 1.3.** Let  $A, B \in GL^+(2)$ . We call transition tilt between  $A$  and  $B$  the absolute tilt of  $AB^{-1}$ , i.e.

$$\tau(AB^{-1}).$$

The transition tilt has an agreeable visual interpretation appearing in Figure 1.2. By Formula (1.1) applied to  $AB^{-1}$ , passing from an image  $Bu$  to an image  $Au$  comprises a single non-Euclidean transformation, namely the central tilt matrix  $T_{\tau(AB^{-1})}$  which squeezes the image in the direction of  $x$  after having rotated it. Thus the transition tilt measures the amount of image distortion caused by a change of view angle. We now state and give a brief proof of the formal properties of the transition tilt stated in [MY09].

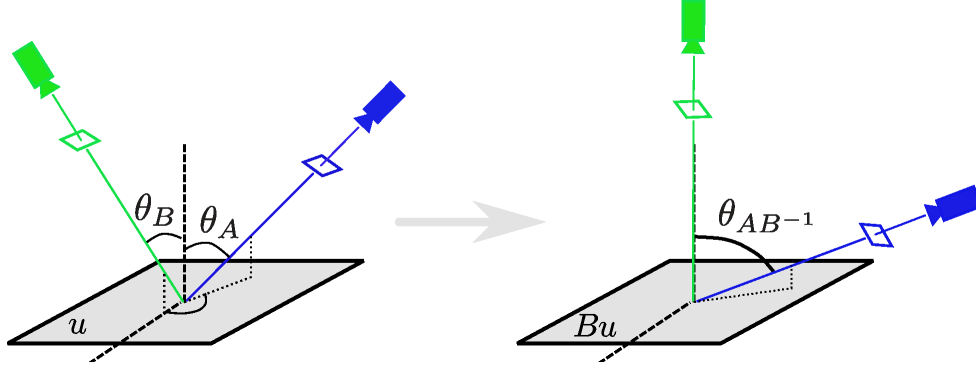


Figure 1.2: Passage from transition tilts (left side) to absolute tilts (right side).

**Proposition 1.3.** For  $A, B \in GL^+(2)$  we have

1.  $\tau(AB^{-1}) = 1 \Leftrightarrow A \sim B$ ;
2.  $\tau(A) = \tau(A^{-1})$ ;
3.  $\tau(AB^{-1}) = \tau(BA^{-1})$ ;
4.  $\tau(AB^{-1}) \leq \tau(A)\tau(B)$ ;
5.  $\max\left\{\frac{\tau(A)}{\tau(B)}, \frac{\tau(B)}{\tau(A)}\right\} \leq \tau(AB^{-1})$ .

*Proof.* 1)

$$\tau(AB^{-1}) = 1 \Leftrightarrow AB^{-1} = \lambda R \Leftrightarrow A = \lambda RB$$

2) By proposition 1.2

$$\begin{aligned} \tau(A) &= \|A\|_2 \|A^{-1}\|_2 \\ &= \tau(A^{-1}) \end{aligned}$$

3) From 2) we have

$$\begin{aligned} \tau(AB^{-1}) &= \tau((AB^{-1})^{-1}) \\ &= \tau(BA^{-1}) \end{aligned}$$

4) By proposition 1.2

$$\begin{aligned} \tau(AB^{-1}) &= \|AB^{-1}\|_2 \|(AB^{-1})^{-1}\|_2 \\ &\leq \|A\|_2 \|B^{-1}\|_2 \|B\|_2 \|A^{-1}\|_2 \\ &= \tau(A)\tau(B) \end{aligned}$$

5) From 4) we have

$$\begin{aligned} \tau(A) &= \tau(AB^{-1}B) \\ &\leq \tau(AB^{-1})\tau(B) \end{aligned}$$

and the same relation for  $B$ . □

**Definition 1.4.** We call Space of Tilts, denoted by  $\Omega$ , the quotient  $GL^+(2) / \sim$  where the equivalence relation  $\sim$  has been given in Definition 1.2.

This proposition completes Definition 1.2 and clarifies the geometrical interpretation of the space of tilts: an element in the space of tilts represents the set of all the camera spins and zooms associated with a certain tilt in a certain direction.

**Notation 1.1.** Let  $A \in GL^+(2)$ . We denote by  $[A]$  the equivalence class in the space of tilts associated to  $A$  i.e.

$$[A] = \{B \in GL^+(2) \mid A \sim B\}.$$

**Definition 1.5.** We denote by  $i$  the canonical injection from the space of tilts to  $GL^+(2)$  defined by

$$i : \begin{cases} \Omega & \rightarrow GL^+(2) \\ [A] & \mapsto T_{\tau(A)} R_{\phi(A)} \end{cases}.$$

This injection filters out the canonical representative from each class which is a mere tilt in the  $x$  direction.

**Remark 1.4.** Clearly, the function  $i$  satisfies

$$[A] = [i([A])]$$

and the space of tilts can be parameterized by picking these representative elements in each class as

$$\Omega = [Id] \cup \left\{ \bigcup_{(t,\phi) \in ]1,\infty[ \times ]0,\pi[} [T_t R_\phi] \right\}.$$

The next proposition brings an additional justification to Definition 1.4. It means that the transition tilt does not depend on the choice of the class representative in the space of tilts.

**Proposition 1.4.** Let  $A, B, C, D \in GL^+(2)$  satisfying  $C \in [A]$  and  $D \in [B]$ . Then

$$\tau(AB^{-1}) = \tau(CD^{-1}).$$

*Proof.* Let  $C \in [A]$ ,  $D \in [B]$ . We first remark that if either  $A \in GO^+(2)$  or  $B \in GO^+(2)$  then the transition tilt operation is respectively the absolute tilt of  $D$  or  $C$ , which does not depend on the class representative.

So without loss of generality suppose  $A, B \in GL^+_{\ast}(2)$ . Then, by proposition 1.1, they are re-written in a unique way as

$$\begin{aligned} A &= \lambda_A Q_A T_s R_A \\ B &= \lambda_B Q_B T_t R_B \end{aligned}$$

and the same result can be applied to the following two matrices

$$\begin{aligned} AB^{-1} &= \lambda_{AB^{-1}} Q_{AB^{-1}} T_{\tau(AB^{-1})} R_{AB^{-1}} \\ T_s R_A R_B^{-1} T_t^{-1} &= \alpha Q_3 T_{t_3} R_3. \end{aligned} \tag{1.2}$$

Moreover

$$\begin{aligned} AB^{-1} &= \lambda_A Q_A T_s R_A (\lambda_B Q_B T_t R_B)^{-1} \\ &= \frac{\alpha \lambda_A}{\lambda_B} \underbrace{(Q_A Q_3)}_{\text{rotation}} T_{t_3} \underbrace{(R_3 Q_B^{-1})}_{\text{rotation}}. \end{aligned}$$

Then, by uniqueness of decomposition in equation (1.2) we have  $T_{\tau(AB^{-1})} = T_{t_3}$ , implying

$$\tau(AB^{-1}) = \tau(T_s R_A R_B^{-1} T_t^{-1}).$$

Again, the same methodology applied to

$$\begin{aligned} C &= \lambda_C Q_C A \\ &= \lambda_C \lambda_A Q_C Q_A T_s R_A \end{aligned}$$

and

$$\begin{aligned} D &= \lambda_D Q_D B \\ &= \lambda_D \lambda_B Q_D Q_B T_t R_B \end{aligned}$$

shows that

$$\tau(CD^{-1}) = \tau(T_s R_A R_B^{-1} T_t^{-1}) = \tau(AB^{-1}).$$

□

The next proposition follows directly from Proposition 1.3.

**Proposition 1.5.** *The function  $d$*

$$d : \begin{cases} \Omega \times \Omega & \rightarrow \mathbb{R}_+ \\ ([A], [B]) & \mapsto \log \tau(AB^{-1}) \end{cases}$$

*is a metric acting on the space of tilts.*

*Proof.* First,  $d$  is well defined thanks to Proposition 1.4 which ensures the independence from class representatives. Let us now prove the four metric axioms:

1) By definition of the absolute tilt  $\forall A, B \in GL^+(2)$  one has that  $\tau(AB^{-1}) \geq 1$ . This implies

$$d([A], [B]) \geq 0.$$

2) By Proposition 1.3-1)  $\forall A, B \in GL^+(2)$

$$\begin{aligned} d([A], [B]) = 0 &\Leftrightarrow \tau(AB^{-1}) = 1 \\ &\Leftrightarrow A \sim B \\ &\Leftrightarrow [A] = [B] \end{aligned}$$

3)  $\forall A, B \in GL^+(2)$ , Proposition 1.3-3) states that

$$\tau(AB^{-1}) = \tau(BA^{-1})$$

which implies

$$d([A], [B]) = d([B], [A])$$

4)  $\forall A, B, C \in GL^+(2)$ , Proposition 1.3-4) assures that the following inequality holds

$$\tau\left(BC^{-1}\left(AC^{-1}\right)^{-1}\right) \leq \tau\left(BC^{-1}\right)\tau\left(AC^{-1}\right).$$

As the logarithm is monotone in  $[1, \infty[$ , by simply applying it to both sides one obtains the triangular inequality for  $d$ . □

### 1.2.3 Neighborhoods in the space of tilts

Now that we have introduced the space of tilts and the adequate metric on this space to measure image distortion, we wish to explore optimal coverings for this space. We start by establishing closed formulas for disks in this 2D space.

**Theorem 1.1.** *Given an element of the space of tilts in canonical form  $[T_t R(\phi_1)]$ , the disk  $\mathcal{B}([T_t R(\phi_1)], r)$  in the space of tilts centered at this element and with radius  $r$  corresponds to the following set*

$$\left\{ [T_s R(\phi_2)] \mid G(t, s, \phi_1, \phi_2) \leq \frac{e^{2r} + 1}{2e^r} \right\}$$

where

$$G(t, s, \phi_1, \phi_2) = \left( \frac{\frac{t}{s} + \frac{s}{t}}{2} \right) \cos^2(\phi_1 - \phi_2) + \left( \frac{\frac{1}{st} + st}{2} \right) \sin^2(\phi_1 - \phi_2).$$

The proof of this theorem is given in the appendix. Figure 1.3 displays such disks in polar coordinates  $(\log \tau \cos(\phi), \log \tau \sin(\phi))$ . This representation will be convenient to visualize region coverings defined by disks in the space of tilts. Figure 1.4 is illustrating an observation hemisphere, which displays in a geometric environment the space of tilts, the class of affine transformations in question (green dots) and their neighborhoods (black surfaces). Notice that green dots represent camera viewpoints as depicted in Figure 1.1. In both representations, the pairs  $(\tau, \phi)$  and  $(\tau, \phi + \pi)$  are denoting the same element of the space of tilts. This is easily interpreted: Two identical images of a planar scene are indeed obtained by an affine camera positioned with a  $\pi$  longitude difference.

**Proposition 1.6.** *Let  $A, B, C \in GL^+(2)$ . Then*

$$[A]C = [AC],$$

i.e., classes in  $\Omega$  are stable by right multiplication. Moreover,

$$d([AC], [BC]) = d([A], [B]).$$

*Proof.* 1) Proof of  $[A]C \subset [AC]$ .

$$\begin{aligned} B \in [A] &\implies B = \lambda R A \\ &\implies BC = \lambda R AC \\ &\implies BC \in [AC] \end{aligned}$$

2) Proof of  $[AC] \subset [A]C$ .

$$\begin{aligned} D \in [AC] &\implies D = \lambda R AC \\ &\implies D \in [A]C \end{aligned}$$

3)

$$\begin{aligned} d([AC], [BC]) &= \log \tau (AC (BC)^{-1}) \\ &= \log \tau (AB^{-1}) \\ &= d(A, B) \end{aligned}$$

□

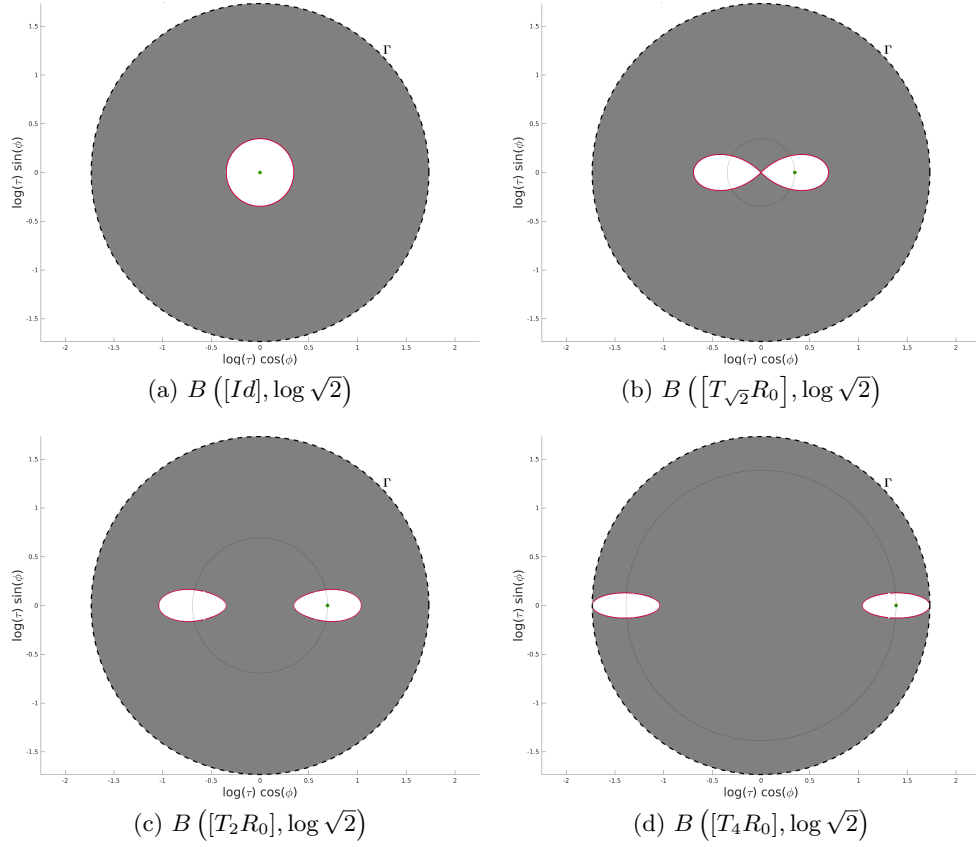


Figure 1.3: (Polar coordinates)

Green point - Affine transformation in question  
Dashed line -  $\partial B([Id], \log 4\sqrt{2})$   
Dotted line - Equal tilts  
Red line - Disk's boundary

**Remark 1.5.** Proposition 1.6 guarantees that transition tilts remain unchanged by right compositions. Furthermore, as argued in the proof of Proposition 1.7, the right composition with an element  $C \in GL^+(2)$  could be seen as a modification from a hypothetical frontal image  $u$  to another hypothetical frontal image  $C^{-1}u$ . All this gives both motivation and meaning to the forthcoming Theorem 1.2.

**Remark 1.6.** One might also be interested in the way disks are transformed by left multiplication of elements belonging to  $GL^+(2)$ . Unfortunately, in general

$$C[A] \neq [CA].$$

Take for example  $C = A = T_t$  so

$$R_{\frac{\pi}{2}} = T_t \left( \frac{1}{t} R_{\frac{\pi}{2}} T_t \right) \notin [T_{t^2}].$$



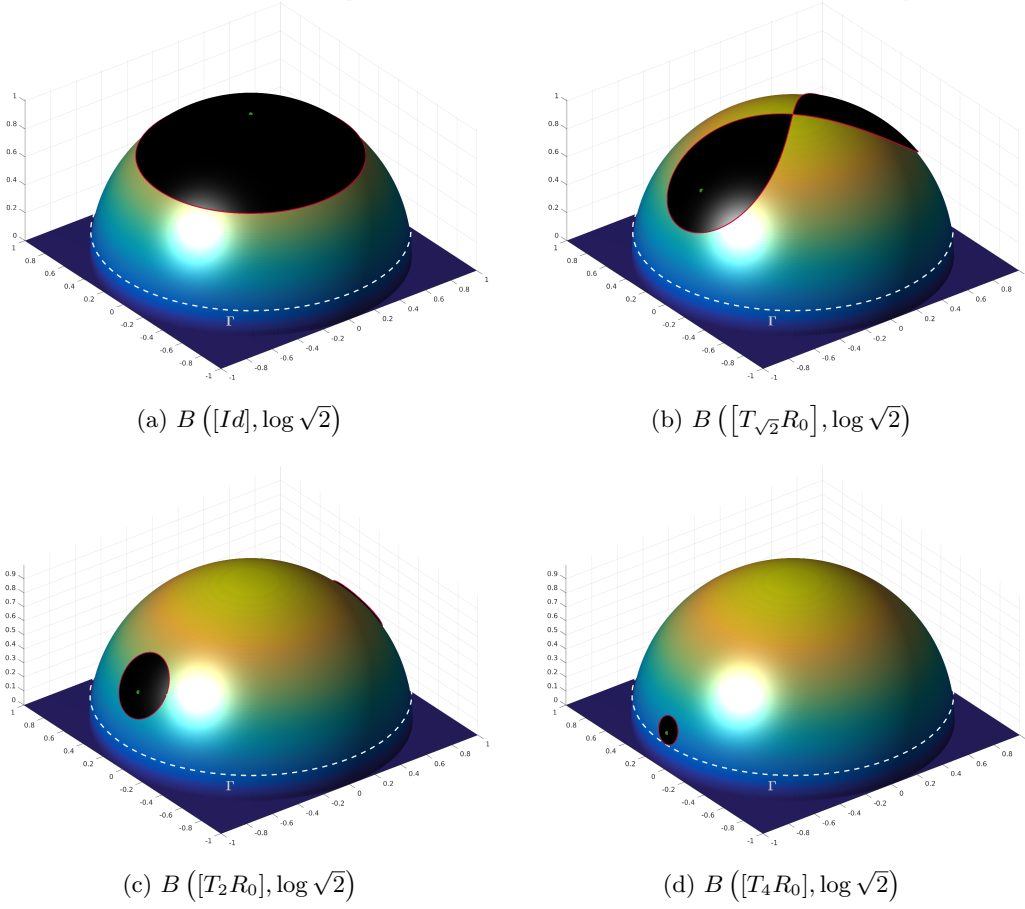


Figure 1.4: (Perspective views)

Green point - Affine transformation in question  
Dashed line -  $\partial B([Id], \log 4\sqrt{2})$   
Black surface - Disk in question

Furthermore, for  $C \in GL^+(2)$  one has

$$\begin{aligned}
\tau(CAB^{-1}C^{-1}) &= c_2(CAB^{-1}C^{-1}) \\
&= \|CAB^{-1}C^{-1}\|_2 \|C(AB^{-1})^{-1}C^{-1}\|_2 \\
&\leq \|C\|_2^2 \|C^{-1}\|_2^2 \|AB^{-1}\|_2 \|(AB^{-1})^{-1}\|_2 \\
&= \tau(C)^2 \tau(AB^{-1})
\end{aligned}$$

so, in general

$$d([CA], [CB]) \leq 2d([C], [Id]) + d([A], [B]).$$

The following theorem will be crucial in the next Section to explain why IMAS algorithms are truly affine invariant.

**Theorem 1.2.** *Let*

$$\begin{aligned}
\Gamma_1 &= \mathcal{B}([Id], \log \Lambda_1) \\
\Gamma_2 &= \mathcal{B}([Id], \log \Lambda_2) \\
\Gamma' &= \mathcal{B}([Id], \log \Lambda_2 r).
\end{aligned}$$

be three neighborhoods of  $[Id]$  in  $\Omega$  where  $\Lambda_1, \Lambda_2, r \in [1, \infty[$ , and assume that  $\mathbb{S}_1, \mathbb{S}_2 \subset \Omega$  are two  $\log r$ -coverings of  $\Gamma_1$  and  $\Gamma'$ , i.e

$$\begin{aligned}\Gamma_1 &\subset \bigcup_{S \in \mathbb{S}_1} \mathcal{B}(S, \log r) \\ \Gamma' &\subset \bigcup_{S \in \mathbb{S}_2} \mathcal{B}(S, \log r).\end{aligned}$$

Then, for every  $[A] \in \Gamma_1$ ,  $[B] \in \Gamma_2$ , there exist  $C \in GL^+(2)$  with  $\tau(C) \leq r$ ,  $S_A \in \mathbb{S}_1$  and  $S_B \in \mathbb{S}_2$  such that

$$\begin{aligned}d\left(S_A, \left[(AC)^{-1}\right]\right) &= 0 \\ d\left(S_B, \left[(BC)^{-1}\right]\right) &\leq \log r.\end{aligned}$$

A sketch of Theorem 1.2 appears in Figure 1.5.

*Proof.* Let us set  $C = A^{-1}i(S_A)^{-1}$  where  $i$  appears in Definition 1.5.

1) Proof of  $d\left(S_A, \left[(AC)^{-1}\right]\right) = 0$ .

Proposition 1.3-2) directly implies

$$d([Id], [A]) = d([Id], [A^{-1}]).$$

Then, as  $\mathbb{S}_1$  is a  $\log r$ -covering of  $\Gamma_1$ , there exists  $S_A \in \mathbb{S}_1$  such that

$$[A^{-1}] \in \mathcal{B}(S_A, \log r)$$

meaning that, the following inequality holds

$$\begin{aligned}d\left([Id], \left[A^{-1}i(S_A)^{-1}\right]\right) &= \log \tau\left(A^{-1}i(S_A)^{-1}\right) \\ &= d\left([A^{-1}], S_A\right) \\ &\leq \log r.\end{aligned}$$

Finally, as  $d$  is a metric (by Proposition 1.5) we know

$$d\left(S_A, \left[(AC)^{-1}\right]\right) = d\left(S_A, [i(S_A)]\right) = 0.$$

2) Proof of  $d\left(S_B, \left[(BC)^{-1}\right]\right) \leq \log r$ .

By first using Proposition 1.3 followed by Proposition 1.5 we have

$$\begin{aligned}\tau(BC) &\leq \tau(B) \tau(C^{-1}) = \Lambda_2 r \\ &\Downarrow \\ d\left([Id], \left[(BC)^{-1}\right]\right) &= \log \tau(BC) \leq \log \Lambda_2 r \\ &\Downarrow \\ \left[(BC)^{-1}\right] &\in \Gamma' .\end{aligned}$$

Once more, as  $\mathbb{S}_2$  is a  $\log r$ -covering of  $\Gamma'$ , there exists  $S_B \in \mathbb{S}_2$  such that

$$\left[(BC)^{-1}\right] \in \mathcal{B}(S_B, \log r).$$

□

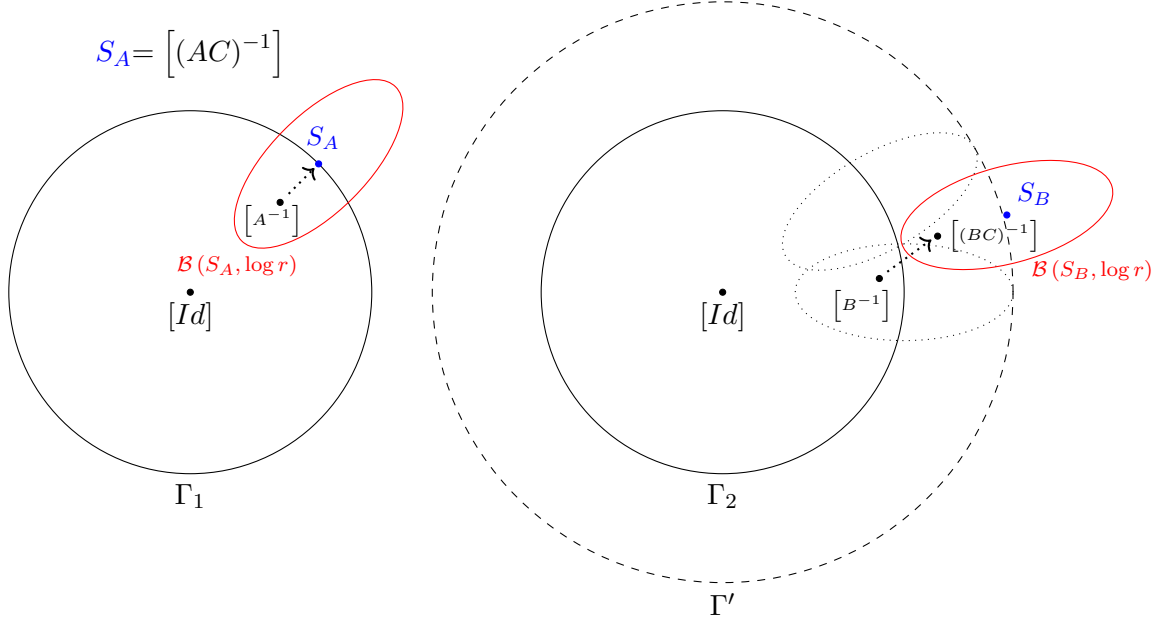


Figure 1.5: Sketch of Theorem 1.2.

### 1.3 Application: optimal affine invariant image matching algorithms

The theory and results presented above provide a well suited geometrical framework for image matching by affine simulation (IMAS). This section gives the mathematical formalism and a mathematical proof that IMAS based algorithms are fully affine invariant, up to sampling errors. While the former sections only dealt with affine geometry, we now must introduce in the formalism the camera blur, as we shall deal with digital image recognition. Our goal is to define rigorously affine invariant recognition for digital images.

Consider a continuous and bounded image  $u(\mathbf{x})$  defined for every  $\mathbf{x} = (x, y) \in \mathbb{R}^2$ . All continuous image operators including the sampling will be written in capital letters  $A, B$  and their composition as a mere juxtaposition  $AB$ .

**Definition 1.6.** For any  $A \in GL^+(2)$ , we define the affine transform  $A$  of a continuous image  $u$  by

$$Au(\mathbf{x}) := u(A\mathbf{x}).$$

Homotheties and rotations acting on continuous images are similarly written as

$$\begin{aligned} H_\lambda u(\mathbf{x}) &= u(\lambda\mathbf{x}); \\ R_\phi u(\mathbf{x}) &= u(R_\phi\mathbf{x}). \end{aligned}$$

We now introduce a compact notation for the various convolutions with Gaussians. We shall denote by  $\star_x$  the 1-D convolution operator in the  $x$ -direction, i.e.

$$G \star_x u(x, y) = \int_{\mathbb{R}} G(z) u(x - z, y) dz.$$

Similarly, we denote by  $\star_y$  the 1-D convolution operator in the  $y$ -direction. We denote by  $\mathbb{G}_\sigma$ ,  $\mathbb{G}_\sigma^x$  and  $\mathbb{G}_\sigma^y$  respectively the 2D and 1D convolution operators in the  $x$  and  $y$

directions with

$$\begin{aligned} G_{\mathbf{c}\sigma}(x, y) &:= \frac{1}{2\pi(\mathbf{c}\sigma)^2} e^{-\frac{x^2+y^2}{2(\mathbf{c}\sigma)^2}} \\ G_{\mathbf{c}\sigma}^x(x) &:= \frac{1}{\sqrt{2\pi\mathbf{c}\sigma}} e^{-\frac{x^2}{2(\mathbf{c}\sigma)^2}} \\ G_{\mathbf{c}\sigma}^y(y) &:= \frac{1}{\sqrt{2\pi\mathbf{c}\sigma}} e^{-\frac{y^2}{2(\mathbf{c}\sigma)^2}} \end{aligned}$$

namely

$$\begin{aligned} \mathbb{G}_\sigma u &:= G_{\mathbf{c}\sigma} \star u \\ \mathbb{G}_\sigma^x u &:= G_{\mathbf{c}\sigma}^x \star_x u \\ \mathbb{G}_\sigma^y u &:= G_{\mathbf{c}\sigma}^y \star_y u. \end{aligned}$$

Here the constant  $c \geq 0.7$  is large enough to ensure that all convolved images, initially sampled at 1 distance, can be sub-sampled at Nyquist distance  $\sigma$  without causing significant aliasing.

**Remark 1.7.**  $\mathbb{G}_\sigma$  satisfies the semigroup property

$$\mathbb{G}_\sigma \mathbb{G}_\beta = \mathbb{G}_{\sqrt{\sigma^2 + \beta^2}}. \quad (1.3)$$

By a mere change of variables in the integral defining the convolution, the next formula holds and will be useful:

$$\mathbb{G}_\sigma H_\gamma u = H_\gamma \mathbb{G}_{\sigma\gamma} u. \quad (1.4)$$

In the classic Shannon-Nyquist framework, we shall denote the image sampling operator (on a unary grid) by  $\mathbf{S}_1$ . Thus  $\mathbf{S}_1 u$  is defined on the grid  $\mathbb{Z}^2$ . The Shannon-Whittaker interpolator of a digital image on  $\mathbb{Z}^2$  will be denoted by  $I$ .

As developed in [YM11], the whole image comparison process, based on local features, can proceed as though images were (locally) obtained by using digital cameras that stand far away, at infinity. The geometric deformations induced by the motion of such cameras are affine maps. A model is also needed for the two main camera parameters not deducible from its position, namely sampling and blur. The digital image is defined on the camera CCD plane. The pixel width can be taken as length unit, and the origin and axes chosen so that the camera pixels are indexed by  $\mathbb{Z}^2$ . The digital initial image is always assumed well-sampled and obtained by a Gaussian blur with standard deviation around 0.8. In all that follows,  $u_0$  denotes the (theoretical) infinite resolution image that would be obtained by a frontal snapshot of a plane object with infinitely many pixels. The digital image obtained by any camera at infinity is therefore formalized as  $\mathbf{u} = \mathbf{S}_1 \mathbb{G}_1 A \mathcal{T} u_0$ , where  $A$  is any linear map with positive singular values and  $\mathcal{T}$  any plane translation. Thus we can summarize the general image formation model with cameras at infinity as follows.

**Definition 1.7 (Image formation model).** *Digital images of a planar object whose frontal infinite resolution image is  $u_0$ , obtained by a digital camera far away from the object, satisfy*

$$\mathbf{u} =: \mathbf{S}_1 \mathbb{G}_1 A \mathcal{T} u_0 \quad (1.5)$$

where  $A$  is any linear map and  $\mathcal{T}$  any plane translation.  $\mathbb{G}_1$  denotes a Gaussian kernel broad enough to ensure no aliasing by 1-sampling, namely  $I \mathbf{S}_1 \mathbb{G}_1 A \mathcal{T} u_0 = \mathbb{G}_1 A \mathcal{T} u_0$ .

The image formation model in Definition 1.7 is illustrated in Figure 1.6.

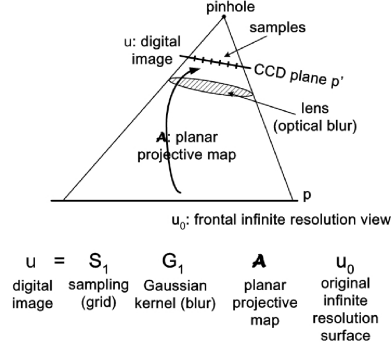


Figure 1.6: The projective camera model  $u = S_1 G_1 A u_0$ .  $A$  is a planar projective transform (a homography).  $G_1$  is an anti-aliasing Gaussian filtering.  $S_1$  is the CCD sampling.

### 1.3.1 Inverting tilts

We now formalize the notion of tilt. There are actually three different notions of tilt, that we must carefully distinguish.

**Definition 1.8.** Given  $t > 1$ , the tilt factor, define:

- Geometric tilts

$$T_t^x u_0(x, y) =: u_0(tx, y);$$

$$T_t^y u_0(x, y) =: u_0(x, ty).$$

- Simulated tilts (taking into account camera blur)

$$\mathbb{T}_t^x v =: T_t^x G_{\sqrt{t^2-1}}^x \star_x v;$$

$$\mathbb{T}_t^y v =: T_t^y G_{\sqrt{t^2-1}}^y \star_y v.$$

- Digital tilts (transforming a digital image  $u$  into a digital image)

$$u \rightarrow S_1 \mathbb{T}_t^x I u;$$

$$u \rightarrow S_1 \mathbb{T}_t^y I u.$$

Digital tilts are the ones used in practice. It all adds up because the simulated tilt yields a blur permitting  $S_1$ -sampling.

If  $u_0$  is an infinite resolution image observed with a camera tilt of  $t$  in the  $x$  direction, the observed image is  $G_1 T_t^x u_0$ . Our main problem is to reverse such tilts. This operation is in principle impossible, because geometric tilts do not commute with blur. However, the first formula of the next Theorem 1.3 shows that  $\mathbb{T}_t^y$  is, up to a zoom out, a pseudo inverse to  $T_t^x$ .

The meaning of this result is that a tilted image  $G_1 T_t^x u$  can be tilted back by tilting in the orthogonal direction. The price to pay is a  $t$  zoom out. The second relation in the theorem means that the application of the simulated tilt to an image that can be well sampled by  $S_1$  yields an image that keeps that well sampling property.

**Theorem 1.3.** Let  $t \geq 1$ . Then

$$\mathbb{T}_t^y G_1 T_t^x = G_1 H_t; \tag{1.6}$$

$$\mathbb{T}_t^y G_1 = G_1 T_t^y. \tag{1.7}$$

*Proof.* Since  $H_t = T_t^y T_t^x$ , (1.6) follows from (1.7) by composing both sides on the right by  $T_t^x$ . Let us now prove (1.7). We shall use the following obvious facts

$$\mathbb{G}_1 = \mathbb{G}_1^x \mathbb{G}_1^y = \mathbb{G}_1^y \mathbb{G}_1^x \quad (1.8)$$

which follows from the separability of the Gaussian and Fubini's theorem and the commutation

$$\mathbb{G}_1^x T_t^y = T_t^y \mathbb{G}_1^x \quad (1.9)$$

which is true because  $\mathbb{G}_1^x$  and  $T_t^y$  act separably on the variables  $x$  and  $y$ . Using first (1.4) in the  $y$  dimension where  $T_t^y$  is a mere homothety, and then successively (1.9), (1.8), the semigroup property for the Gaussians, and Definition 1.8 we get

$$\begin{aligned} T_t^y \mathbb{G}_t^y &= \mathbb{G}_1^y T_t^y \Rightarrow \\ \mathbb{G}_1^x T_t^y \mathbb{G}_t^y &= \mathbb{G}_1^x \mathbb{G}_1^y T_t^y \Rightarrow \\ T_t^y \mathbb{G}_t^y \mathbb{G}_1^x &= \mathbb{G}_1 T_t^y \Rightarrow \\ T_t^y \mathbb{G}_{\sqrt{t^2-1}}^y \mathbb{G}_1^y \mathbb{G}_1^x &= \mathbb{G}_1 T_t^y \Rightarrow \\ \mathbb{T}_t^y \mathbb{G}_1 &= \mathbb{G}_1 T_t^y, \end{aligned}$$

which proves (1.7). □

The meaning of Theorem 1.3 is that we can design an exact algorithm that simulates all inverse tilts for comparing two digital images. This algorithm handles two images  $u = \mathbb{G}_1 A \mathcal{T}_1 w_0$  and  $v = \mathbb{G}_1 B \mathcal{T}_2 w_0$  that are two snapshots from different view points of a flat object whose front infinite resolution image is denoted by  $w_0$ .

### 1.3.2 Proof that IMAS works

In this section, the formal IMAS algorithm is duly presented (Algorithm 1). Our goal is to prove that it works. This proof is a direct application of the results introduced of the previous section. The algorithm and its proof rely on the formal assumption that there exists an image comparison algorithm able to compare image pairs with tilts lower than  $r$ . The core idea of IMAS algorithms is illustrated in Figure 1.7.

---

**Algorithm 1** Formal IMAS (Image Matching by Affine Simulation)

---

**environment:**

Parameters and assumptions from Theorem 1.2 with

$$\mathbb{S}_i = \left\{ \left[ T_{t_k^i}^x R_{\phi_k^i} \right] \right\}_{k=1, \dots, n_i}.$$

**input:**Query and target images:  $u$  and  $v$ .**start:****foreach**  $k \in \{1, \dots, n_1\}$  **do**

|

$$u_k = \mathbb{T}_{t_k^1}^x R_{\phi_k^1} u.$$

**foreach**  $k \in \{1, \dots, n_2\}$  **do**

|

$$v_k = \mathbb{T}_{t_k^2}^x R_{\phi_k^2} v.$$

**foreach**  $(k_1, k_2) \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}$  **do**

|

$$M_{k_1, k_2} = \text{SIIM-Matches}(u_{k_1}, v_{k_2}).$$

**return:**

$$M = \bigcup_{(k_1, k_2) \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}} M_{k_1, k_2}.$$

---

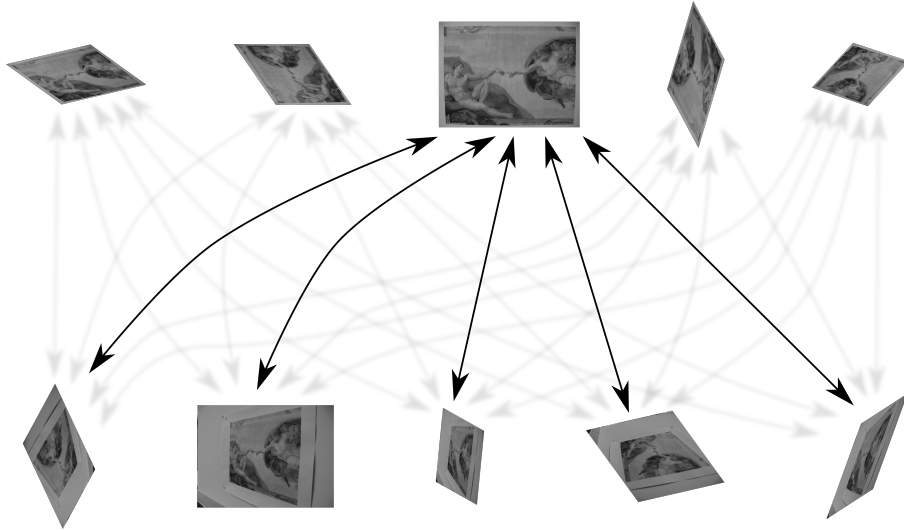


Figure 1.7: IMAS algorithms start by applying a finite set of optical affine simulations to  $u$  and  $v$ , followed by pairwise comparisons.

**Proposition 1.7.** *Let  $u$  and  $v$  be respectively query and target images which are related by a transition tilt under  $\Lambda_1\Lambda_2$ , i.e. there exist a continuous image  $w_0$  and  $A, B \in GL^+(2)$  with*

$$\tau(A) \leq \Lambda_1 \text{ and } \tau(B) \leq \Lambda_2$$

*such that*

$$u = \mathbb{G}_1 A \mathcal{T}_1 w_0 \text{ and } v = \mathbb{G}_1 B \mathcal{T}_2 w_0 \quad (1.10)$$

*where  $\mathcal{T}_1, \mathcal{T}_2$  are planar translations. Then, under the assumptions of Theorem 1.2, the formal IMAS of Algorithm 1 generates two affine versions of the images  $u$  and  $v$  with a transition tilt lower than  $r$ .*

*Proof.* By Theorem 1.2 there exist  $S_A \in \mathbb{S}_1$ ,  $S_B \in \mathbb{S}_2$  and  $C \in GL^+(2)$  with  $\tau(C) \leq r$  such that

$$\begin{aligned} d(S_A, [(AC)^{-1}]) &= 0 \\ d(S_B, [(BC)^{-1}]) &\leq \log r. \end{aligned}$$

Consider the slanted view of the frontal continuous image  $w_0$  defined by  $w_1 := C^{-1}w_0$ . Then we can rewrite query and target images as

$$u = \mathbb{G}_1 A C \mathcal{T}_1 w_1 \text{ and } v = \mathbb{G}_1 B C \mathcal{T}_2 w_1.$$

By Proposition 1.6, the above modification keeps transitions tilts stable, i.e.

$$d([AC], [BC]) = d([A], [B]),$$

so we can reason as if  $w_1$  were the frontal image, instead of  $w_0$ .

Now, the formal IMAS Algorithm 1 will apply  $i(S_A) = T_{t_A}^x R_{\phi_A}$  and  $i(S_B) = T_{t_B}^x R_{\phi_B}$  respectively on the query and target images. This is:

1.  $T_{t_A}^x R_{\phi_A}$  to  $u$ , which yields

$$\begin{aligned} \tilde{u} &= \mathbb{G}_1 i(S_A) A C \mathcal{T}_1 w_1 \\ &= \mathbb{G}_1 \lambda R \mathcal{T}_1 w_1. \end{aligned}$$

2.  $T_{t_B}^x R_{\phi_B}$  to  $v$ , which yields

$$\tilde{v} = \mathbb{G}_1 i(S_B) B C \mathcal{T}_2 w_1.$$

But

$$\begin{aligned} d([Id], [i(S_B) BC]) &= \log \tau(i(S_B) BC) \\ &= d(S_B, [(BC)^{-1}]) \\ &\leq \log r \end{aligned}$$

which proves that the affine relation between  $\tilde{u}$  and  $\tilde{v}$  involves a transition tilt under  $r$ .  $\square$

**Remark 1.8.** *Two  $\log r$ -coverings of the same region*

$$\Gamma = \mathcal{B}([Id], \log \Lambda)$$

*would then ensure that the formal IMAS Algorithm 1 manages to reduce transition tilts under  $\frac{\Lambda^2}{r}$  between two images into transition tilts under  $r$ . A relation between covered absolute tilts, attainable transition tilts and maximal viewpoint angle can be found in Table 1.1.*



**Table 1.1** Link between absolute tilts, transition tilts and viewpoint.

Covered absolute tilts $(\tau(A) \leq \sqrt{r}\Lambda \text{ and } \tau(B) \leq \sqrt{r}\Lambda)$	Attainable transition tilts $(\tau(AB^{-1}) \leq \Lambda^2)$	Viewpoint angle $(\arccos \frac{1}{\Lambda^2})$
$\Lambda = 8$	64	$89.1^\circ$
$\Lambda = 4\sqrt{2}$	32	$88.2^\circ$
$\Lambda = 4$	16	$86.4^\circ$
$\Lambda = 2\sqrt{2}$	8	$82.8^\circ$
$\Lambda = 2$	4	$75.5^\circ$
$\Lambda = \sqrt{2}$	2	$60^\circ$

### 1.3.3 Optimal discrete coverings in the space of tilts

We now consider the problem of providing two optimal sets  $\mathbb{S}_1, \mathbb{S}_2 \subset \Omega$  permitting the application of Theorem 1.2. These sets should ensure a minimal complexity for the IMAS algorithm. We thus need to define an optimality criterion. We observe that an IMAS algorithm simulates affine transformations on a digital image and then compares descriptors coming from those simulated versions. One would like to minimize the overall number of descriptor comparisons while maintaining the detection efficiency. This minimization *is not* equivalent to a minimization of the number of simulated versions being used. We shall base our efficiency criterion on two straightforward remarks. The first one is that if a digital image suffers a tilt  $t$  in any direction, its area gets modified by a factor  $\frac{1}{t}$ . The second one is that the expected number of keypoints in a digital image is proportional to its area. Both remarks imply that the complexity of an IMAS algorithm will be given by the overall area of the simulated images being ultimately compared. This justifies the next definition.

**Definition 1.9.** We call area ratio of  $\mathbb{S}$  (a finite set of elements in  $\Omega$ ) the real number

$$\sum_{S \in \mathbb{S}} \frac{1}{\tau(S)}.$$

The area ratio fixes the factor (larger than 1) by which the image area is being multiplied when summing the areas of all of its tilted versions. Then, as the ultimate goal is to reduce the number of key points comparisons, it is natural to look for a set  $\mathbb{S}$  whose area ratio is close to the infimum among all  $\log r$ -coverings of  $\Gamma$ . Unfortunately, even in  $\mathbb{R}^2$ , the mathematical problem of finding a covering of a certain set with a minimum amount of disks is well known to be NP-hard. It is therefore difficult to find an optimal solution for our problem, and unlikely that it will be proved to be optimal even if it is. Fortunately, our search space in the set of  $\log r$ -coverings can be drastically reduced by imposing practical and theoretical constraints to  $\mathbb{S}$ . Those constraints follow from simple requirements for an image matching method.

**Definition 1.10.** We shall say that a set  $\mathbb{S} \in \Omega$  is feasible if and only if:

1.  $[Id] \in \mathbb{S}$ .
2. There exist  $n \in \mathbb{N}^+$  and

$$(t_1, \dots, t_n, \phi_1, \dots, \phi_n) \in [1, \infty[^n \times ]0, \pi]^n$$

such that

$$\mathbb{S} \setminus \{[Id]\} = \bigcup_{i=1}^n \left\{ [T_{t_i} R_{\phi_i}] \in \Omega \mid k = 0, \dots, \left\lfloor \frac{\pi}{\phi_i} \right\rfloor \right\}$$

where  $\lfloor a \rfloor$  denotes the nearest integer less than or equal to a real number  $a$ .

**Remark 1.9.** Definition 1.10-1) avoids an image resolution loss before comparison, an obvious requirement. Imposing groups of concentric equidistant tilts as in Definition 1.10-2) is a sound isotropy requirement.

**Definition 1.11.** Set  $\Gamma = \mathcal{B}([Id], \log \Lambda)$ . A feasible set  $\mathbb{S} \in \Omega$  with parameters

$$(n, (t_1, \dots, t_n, \phi_1, \dots, \phi_n)) \in \mathbb{N}^+ \times [1, \infty[^n \times ]0, \pi]^n$$

is said to be optimal among feasible sets if and only if it realizes the minimal area ratio. In other words, optimal feasible sets are solutions of:

$$\begin{aligned} & \arg \min_{(n, (t_1, \dots, t_n, \phi_1, \dots, \phi_n)) \in \mathbb{N}^+ \times [1, \infty[^n \times ]0, \pi]^n} 1 + \sum_{i=1}^n \frac{|J_{t_i, \phi_i}|}{t_i} \\ & \text{subject to: } \Gamma \subset \mathcal{B}_{[Id]}^{\log r} \cup \left\{ \bigcup_{1 \leq i \leq n} \bigcup_{S \in J_{t_i, \phi_i}} \mathcal{B}_{[S]}^{\log r} \right\} \end{aligned} \quad (1.11)$$

where  $J_{t_i, \phi_i}$  is the set of transformations of the form

$$T_{t_i} R_{\phi_i}, T_{t_i} R_{2\phi_i}, \dots, T_{t_i} R_{\left\lfloor \frac{\pi}{\phi_i} \right\rfloor \phi_i},$$

$|J_{t_i, \phi_i}|$  is the cardinal of  $J_{t_i, \phi_i}$  and  $\mathcal{B}_{[S]}^{\log r}$  is denoting  $\mathcal{B}([S], \log r)$ .

Fortunately for our problem with the realistic values  $\Lambda = 6$  and  $r = 1.8$ ,  $n = 2$  can be fixed, as easy heuristics indicate that any covering with  $n > 2$  has a far too large area ratio. Thus our optimization in a realistic setting ends up being performed in dimension 4 for sets  $(t_1, t_2, \phi_1, \phi_2)$ . With  $n$  thus fixed the optimization problem in (1.11) can be exhaustively optimized. In this minimization we deal with 4 dimensions and more specifically with  $100^4$  feasible sets by sampling each parameter. This yields an almost exact discrete exhaustive optimization by sampling densely the explored set  $(t_1, t_2, \phi_1, \phi_2)$  with 100 different values for each parameter. The next proposition describes the result of this optimization and verifies that it is indeed feasible.

**Proposition 1.8.** There exists a feasible  $\log 1.8$ -covering, depicted in Figure 1.9c, with area ratio equal to 6.34. It is an approximated solution of the optimization problem in (1.11) for  $\Gamma = \{[T_t R_\phi] \mid t \leq 6\}$ ,  $n = 2$ . Therefore, the infimum area ratio among all  $\log 1.8$ -coverings of  $\{[T_t R_\phi] \mid t \leq 6\}$  is lower than 6.34.

*Proof.* We are dealing with 4 dimensions to minimize and more specifically with  $100^4$  feasible sets. Computing area ratios for each feasible set is straightforward but validating the covering condition is a more involved computational issue. For the sake of clearness, the intersection of disks boundaries, which are composed at most of two elements for non identical disks, shall be denoted by

$$\Sigma_1 = \partial \mathcal{B}_{[T_{t_1}]}^{\log 1.8} \cap \partial \mathcal{B}_{[T_{t_1} R_{\phi_1}]}^{\log 1.8} \quad \Sigma_2 = \partial \mathcal{B}_{[T_{t_2}]}^{\log 1.8} \cap \partial \mathcal{B}_{[T_{t_2} R_{\phi_2}]}^{\log 1.8}$$

and their respective closest and farthest elements will be denoted by

$$\begin{aligned} \min \Sigma_1 &:= \arg \min_{S \in \Sigma_1} d(S, [Id]) & \max \Sigma_1 &:= \arg \max_{S \in \Sigma_1} d(S, [Id]), \\ \min \Sigma_2 &:= \arg \min_{S \in \Sigma_2} d(S, [Id]) & \max \Sigma_2 &:= \arg \max_{S \in \Sigma_2} d(S, [Id]). \end{aligned}$$

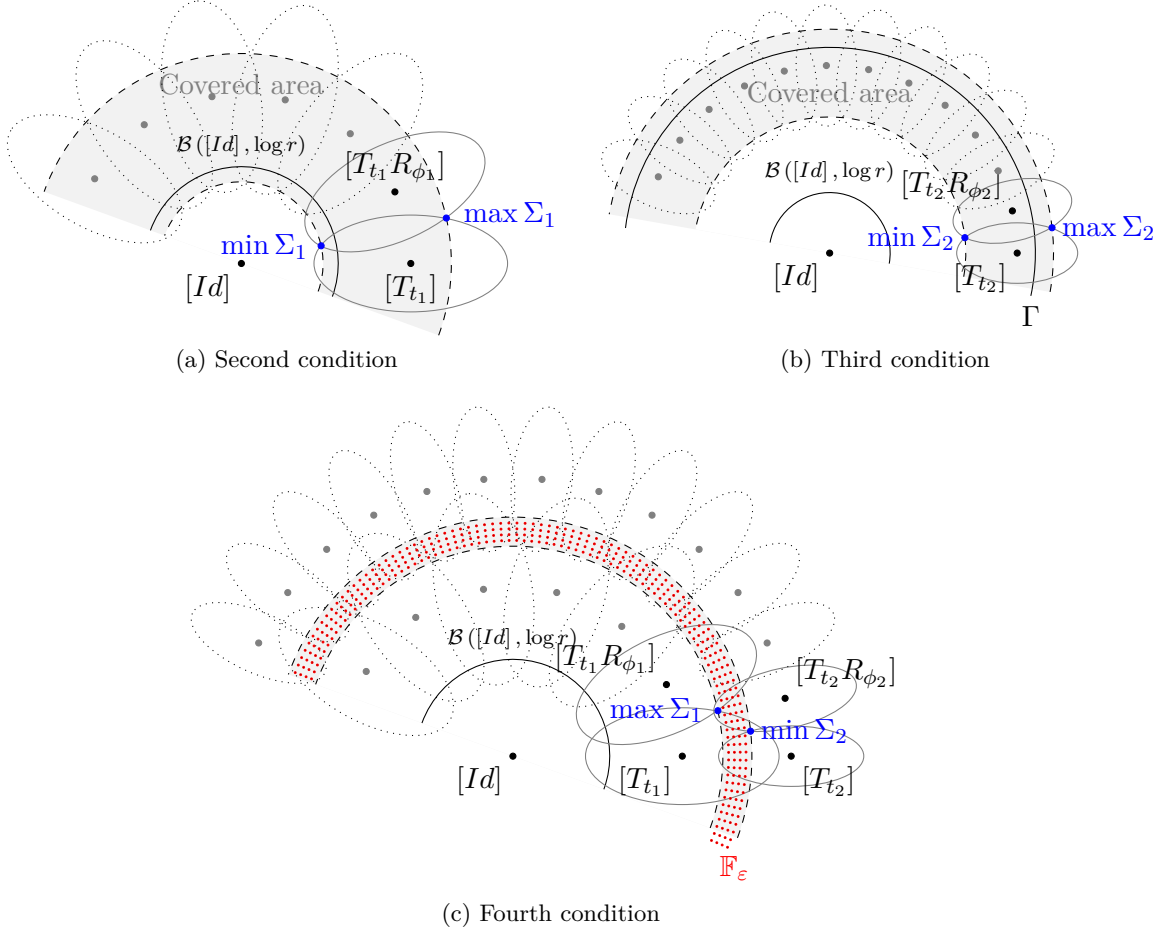


Figure 1.8: Verifying covering conditions for feasible sets in Proposition 1.8.

In order to check if a feasible set does cover the specified region we propose to verify the following four conditions depicted in Figure 1.8:

1.  $\Sigma_1 \neq \emptyset$  and  $\Sigma_2 \neq \emptyset$ .
2.  $\min \Sigma_1$  must lie inside the ball  $\mathcal{B}_{[Id]}^{\log 1.8}$ , which ensures a covering of  $\mathcal{B}_{[Id]}^{\log \tau(\max \Sigma_1)}$ .
3.  $\max \Sigma_2$  must lie outside the region  $\Gamma$ , which ensures a covering of the annulus defined by  $\Gamma \setminus \mathcal{B}_{[Id]}^{\log \tau(\min \Sigma_2)}$ .
4. For  $\varepsilon$  small, all elements  $S \in \mathbb{F}_\varepsilon$  must lie inside some disks of radius  $\log(1.8 - \varepsilon)$ , i.e.

$$S \in \bigcup_{1 \leq i \leq 2} \bigcup_{S' \in J_{t_i, \phi_i}} \mathcal{B}_{[S']}^{\log(1.8 - \varepsilon)},$$

where  $\mathbb{F}_\varepsilon$  is a finite  $\varepsilon$ -dense set of the annulus defined by

$$\mathcal{B}_{[Id]}^{\log \tau(\min \Sigma_2)} \setminus \mathcal{B}_{[Id]}^{\log \tau(\max \Sigma_1)}.$$

Notice that the fourth condition only ensures a  $\log(1.8 - \varepsilon)$ -covering up to an error

$$\varepsilon = \max_{S' \in \Gamma} \min_{S \in \mathbb{F}_\varepsilon} d(S, S')$$

and so, by dilating back disks radius to 1.8 one ensures  $\log 1.8$ -coverings.

By using the procedure described above, an approximated solution to the optimization problem in (1.11) has been obtained. Its parameters can be found in Table 1.2. Its corresponding representation in the space of tilts appears in Figure 1.9c.  $\square$

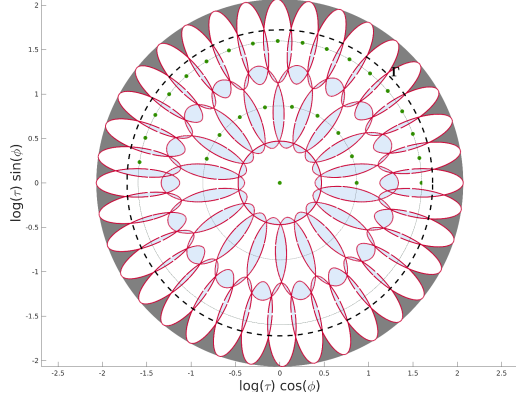
**Table 1.2** Approximated solution to the optimization problem in (1.11)

Parameter	Value
$t_1^{opt}$	2.88447
$\phi_1^{opt}$	0.394085
$t_2^{opt}$	6.2197
$\phi_2^{opt}$	0.196389

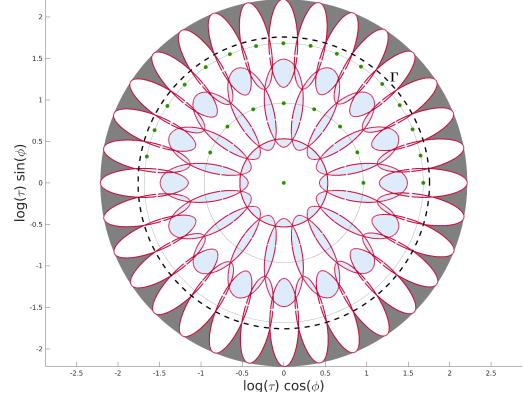
The procedure in the proof of Proposition 1.8 has also been applied to find more near optimal coverings appearing in Figure 1.9.

## 1.4 Experimental Validation

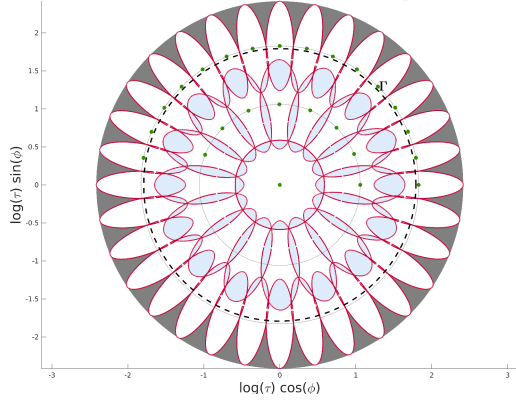
We are now able to propose and evaluate for each SIIM method its IMAS, namely its affine-invariant extension. This affine invariant version relies on two facts. First, each SIIM identifies viewpoint changes, under a certain transition tilt threshold (that we shall estimate in this section). Second, any smooth map is locally approximable by an affine map. Hence, under the assumption that the surface of photographed objects is locally smooth, all viewpoint changes can be understood as local transition tilts changes (see Figure 1.1). Third, once provided with a  $\log r$ -covering of  $\Gamma = \Gamma'$ , where  $r$  is less than the transition tilt threshold of the SIIM, Proposition 1.7 states that Algorithm 1 offers an affine-invariant version of the considered SIIM. Indeed, there is at least one pair of simulated images whose transition tilt is less than  $r$ , and on these two images the SIIM



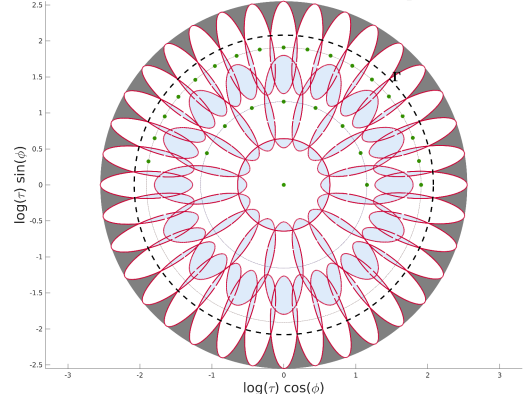
(a) Optimal log 1.6-covering of  $\{[T_t R_\phi] \mid t \leq 5.6\}$  with 28 affine simulations representing an area ratio of 8.42.



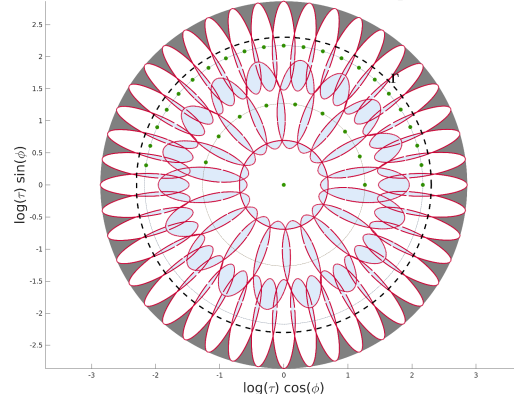
(b) Optimal log 1.7-covering of  $\{[T_t R_\phi] \mid t \leq 5.8\}$  with 25 affine simulations representing an area ratio of 7.06.



(c) Optimal log 1.8-covering of  $\{[T_t R_\phi] \mid t \leq 6\}$  with 25 affine simulations representing an area ratio of 6.34.



(d) Optimal log 1.9-covering of  $\{[T_t R_\phi] \mid t \leq 8\}$  with 27 affine simulations representing an area ratio of 6.18.



(e) Optimal log 2-covering of  $\{[T_t R_\phi] \mid t \leq 10\}$  with 32 affine simulations representing an area ratio of 6.02.

Figure 1.9: Near-optimal coverings in the space of tilts.

Gray areas - Uncovered.

Blue areas - Covered by at least two disks.

White areas - Covered by only one disk.

can succeed. The affine invariance property is ensured for transition tilts changes up to  $\Lambda_1\Lambda_2$ , i.e. for viewpoint angle changes of about  $\arccos\left(\frac{1}{\Lambda_1\Lambda_2}\right)$ . We shall denote by  $t_{\max}^{s_1 \times s_2}$  the associated maximum tilt tolerance with respect to a matching method for images with size larger than  $s_1 \times s_2$ .

In our experiments, all SIIM methods were immersed in the same affine extension set-up. The simulation of optical tilts, matching and filtering were handled in the very same way. This set-up received as a parameter the name of the base detector+extractor method to perform, then a brute force matcher was performed with the second-closest neighbor acceptance criterion proposed by D. Lowe in [Low04]. Finally, as presented in [MY09, YM11], three main filters were applied: first, only unique matches were taken into account; second, groups of multiple-to-one and one-to-multiple matches were removed; finally, only matches coming from the most significant geometric model (if it existed!) were kept. In our case, as all tests were based on planar transformations, the ORSA homography detector [MMM12] (a parameterless variant of RANSAC) was applied to filter out matches not compatible with the dominant homography.

All detectors, all extractors and the matcher were taken from the Open Source Computer Vision (OPENCV) Library, version 3.2.0.

#### 1.4.1 Maximal tilt tolerance computation for each SIIM

From the complexity viewpoint, the main quantitative parameter for extending a SIIM into an IMAS is its tilt tolerance. We do not question the invariance of descriptors with respect to zoom and rotations but rather how they perform against transition tilts changes incurred when matching, for example,  $\mathbb{G}_1 I d u$  to  $\mathbb{G}_1 T_t R_\phi u$  where  $t \in [1, \infty[$  and  $\phi \in [0, \pi[$ .

We used the *tolerance image dataset* displayed in Figure 1.10 to evaluate the maximal tilt tolerance of each SIIM with respect to images of similar size. Images in this dataset have a fixed size and were selected to obtain a diversity of challenging scenarios. In order to approximate  $t_{\max}^{700 \times 550}$ , we simulated optical tilts on the tolerance image dataset and then tested whether this affine simulation was identified by ORSA Homography with a precision of 3 pixels. This test determined upper bounds  $U_{\max}^{700 \times 550}$  depicted in Figure 1.11 for nine of the best state-of-the-art SIIMs.

This test yielded upper bounds for  $t_{\max}^{700 \times 550}$ , based on its application to nine images whose sizes are close to  $700 \times 550$ . Supposing a maximal angle error computation of  $\frac{\pi}{10}$ , we assumed that for each SIIM

$$t_{\max}^{700 \times 550} = \frac{U_{\max}^{700 \times 550}}{\frac{1}{|\cos(\frac{\pi}{10})|}} \approx \frac{U_{\max}^{700 \times 550}}{1.05}$$

and constructed its affine invariant version with  $\log t_{\max}^{700 \times 550}$ -coverings.

#### 1.4.2 Affine-invariant methods

The matching process is as symmetric as possible. No significant changes should come along by interchanging the roles of the query and target images. In the case of IMAS algorithms this symmetry implies a unique set of optical tilts to simulate on both query and target images. Thus, if this unique set of optical tilts represents a  $\log r$ -covering of

$$\Gamma_1 = \Gamma' = \{[T_t R_\phi] \mid t \leq \Lambda\}$$

then Proposition 1.7 ensures that any IMAS based on a SIIM whose maximum tilt tolerance is greater than  $r$  is able to identify all tilts under  $\frac{\Lambda^2}{r}$  by simulating all affine maps in the  $\log r$ -covering.

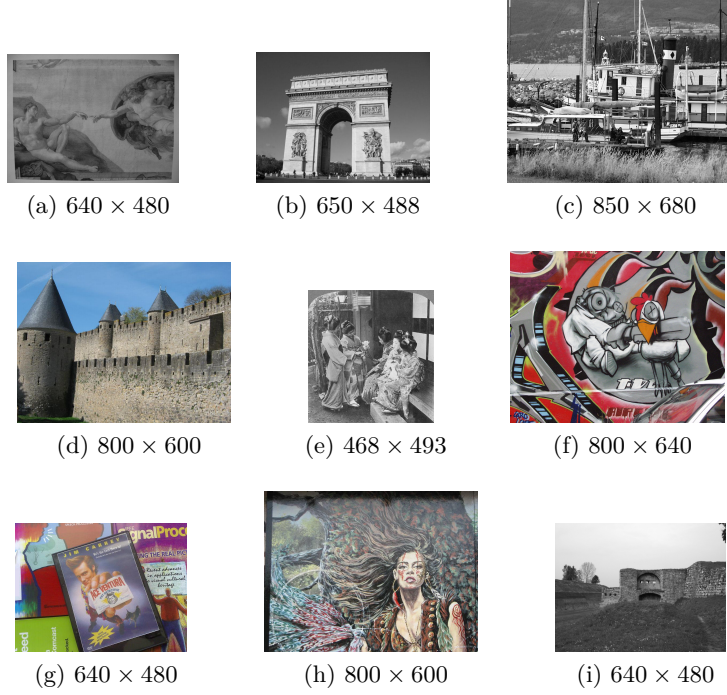


Figure 1.10: Tolerance image dataset.

Several coverings in the space of tilts have been proposed in [MY09, YM11, PLYP12, MMP15] for SIFT and SURF. Figure 1.14 displays these coverings. They are clearly not optimal. Indeed, most of these coverings do not really cover the region they were meant to, except for ASIFT [MY09, YM11] (which instead is visually redundant) and for the affine DoG-SIFT version in [MMP15].

In order to compare the efficiency of those coverings, query and target images were generated in a way so as to test Algorithm 1 to the limit, i.e., forcing the worst case scenario in which  $\left[(BC)^{-1}\right]$  lies in  $\Gamma' \setminus \Gamma_2$ . We simulated the optical tilts on query and target images coming from one single image. This image, denoted by  $w_0$  and appearing in Figure 1.12, was then used to compute the inputs of Algorithm 1 as follows:

- Query image (non-fixed tilt),  $\mathbb{G}_1 A_{t,\phi} w_0$  where  $A_{t,\phi} = R_\phi T_t R_{\frac{\pi}{2}}$ .
- Target image (fixed tilt),  $\mathbb{G}_1 B_\phi w_0$  where  $B_\phi = R_{\phi+\frac{\pi}{2}} T_\Lambda$ .

The veritable interest of these affine maps being the inverse maps they determine, namely,

$$\begin{aligned} \left[A_{t,\phi}^{-1}\right] &= \left[T_t R_{\frac{\pi}{2}-\phi}\right], \\ \left[B_\phi^{-1}\right] &= \left[T_\Lambda R_\phi\right], \end{aligned}$$

which according to Proposition 1.3-4, attain maximal transition tilts for fixed tilts such as  $t$  and  $\Lambda$ , i.e.

$$\tau \left( A_{t,\phi}^{-1} B_\phi \right) = t\Lambda.$$

When ORSA Homography was able to identify the affine map that relates query and target images, we counted the event as a success. Clearly, if  $\Gamma'$  and  $\Gamma_2$  are truly log  $r$ -covered then Proposition 1.7 implies that all tests for which  $\left[A_{t,\phi}^{-1}\right] \in \Gamma_1$  should be counted



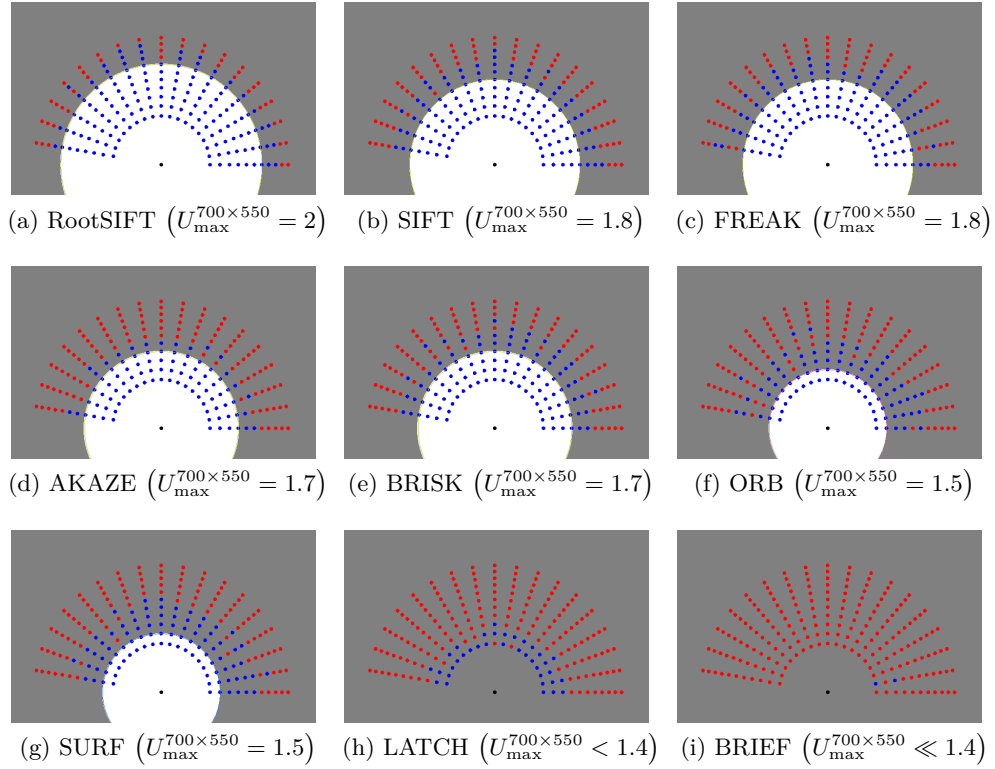


Figure 1.11: Represented in the space of tilts, the associated upper bounds ( $U_{\max}^{700 \times 550}$ ) for maximum tilt tolerances.

Black dot -  $[Id]$ .

Coloured dots stand for tested tilts  $[T_t R_\phi]$  where  $t \in \{1.4, 1.5, \dots, 2.4\}$  and  $\phi \in \{0, 10, \dots, 170\}$ .

Blue dots - attainable tilts for all images in the dataset.

Red dots - unattainable tilts for at least one image in the dataset.

Gray areas -  $\{[T_t R_\phi] \mid t \geq U_{\max}^{700 \times 550}\}$ .

White areas -  $\{[T_t R_\phi] \mid t \leq U_{\max}^{700 \times 550}\}$ .

as a success. Results in Figure 1.13 were as expected and highlight the importance of using the right coverings for extreme cases. Both ASIFT and Optimal Affine-SIFT were able to capt most of all transition tilts that Proposition 1.7 predicted, namely those under  $\frac{\Lambda^2}{r}$ .

We must keep in mind that these log  $r$ -coverings depend on tilt tolerances found over images in Figure 1.10. Maximal tilt tolerances are linked to the size of images being compared and as a consequence the disks radius might grow or shrink proportionally to the minimum size of all simulated images. Moreover, Proposition 1.7 does not take into account discretization errors and relies on two main hypotheses:

1. The considered SIIM is truly rotation and zoom invariant.
2. For images similar to the input image, the SIIM under consideration has a maximal tilt tolerance not smaller than  $r$ .

As anticipated, the area ratio associated to a covering reliably evaluates the difference of performance between affine versions of the same matching method. Being proportionally linked to the total amount of keypoints, the area ratio of Definition 1.9 predicts the order of growth in computation time. For example, the SIFT keypoint computation part induced by the optimal covering in Figure 1.9b is twice faster than the one induced by the





Figure 1.12: Image  $w_0$  ( $3264 \times 1836$ ) for the IMAS efficiency test

ASIFT covering. The same goes for the matching part, only this time the optimal version is four times faster. Since both coverings cover about the same region, our Optimal Affine-SIFT supplants ASIFT with no qualitative matching loss.

Two examples of performance over query and target images from Figure 1.15 and 1.16 are respectively found in Table 1.3 and Table 1.4. In Table 1.3, Affine-ORB and Affine-BRIEF both fail because of too many false matches. The best scores found by ORSA to identify meaningful homographies were respectively 16 out of 905 and 6 out of 1409. Code optimization, smart tweaks and parallelism performance may vary from SIIM to SIIM and from IMAS to IMAS, which ultimately may lead to discrepant area ratio predictions on computation time. This is the case of SURF (and optimal Affine-SURF) whose implementation uses several fine and clever optimizations. Nonetheless, the optimal Affine-SIFT yields more matches for a lower computation time.

In Table 1.4 the reader will notice that Affine-ORB has less matches than ORB itself, which might seem contradictory. This happens when post-processing the matches, more specifically, when applying the second filter. The *multiple-to-one/one-to-multiple* filter, initially proposed in [MY09, YM11], is meant to filter out undesired aberrant matches but, unfortunately, many good ones get also eliminated. In spite of this handicap, Affine-ORB is able to catch more matches with higher transition tilts.

**Table 1.3** Matching methods performance over query and target images from Figure 1.15. The proposed matching methods in this chapter appear in bold. Computations were performed on an Intel(R) Core(TM) i5-4210U CPU 1.70GHz with 2 cores.

M - Matches.

*ar* - area ratio.

	M	<i>ar</i>	$ar^2$	Keypoints (seconds)	Matching (seconds)	Filters (seconds)
SIFT	0	1	1	0.69	0.70	0.18
ASIFT	1013	13.7	189.6	12.46	138.59	3.05
<b>(Optimal) Affine-SIFT</b>	<b>795</b>	<b>7.06</b>	<b>49.8</b>	<b>6.04</b>	<b>29.61</b>	<b>1.39</b>
RootSIFT	0	1	1	0.72	0.71	0.18
<b>Affine-RootSIFT</b>	<b>658</b>	<b>6.9</b>	<b>47.6</b>	<b>5.05</b>	<b>20.70</b>	<b>1.44</b>
SURF	0	1	1	1.01	0.79	0.19
<b>(Optimal) Affine-SURF</b>	<b>471</b>	<b>14.82</b>	<b>219,6</b>	<b>12.53</b>	<b>35.24</b>	<b>1.40</b>
BRISK	0	1	1	1.75	0.27	0.18
<b>Affine-BRISK</b>	<b>421</b>	<b>8.42</b>	<b>70,89</b>	<b>18.95</b>	<b>8.68</b>	<b>2.06</b>
BRIEF	0	1	1	0.05	0.01	0.19
<b>Affine-BRIEF</b>	<b>0</b>	<b>14.82</b>	<b>219,6</b>	<b>4.20</b>	<b>2.18</b>	<b>6.08</b>
ORB	0	1	1	0.05	0.02	0.17
<b>Affine-ORB</b>	<b>0</b>	<b>14.82</b>	<b>219,6</b>	<b>4.34</b>	<b>5.13</b>	<b>3.25</b>
AKAZE	0	1	1	0.42	0.13	0.21
<b>Affine-AKAZE</b>	<b>194</b>	<b>8.42</b>	<b>70,89</b>	<b>5.00</b>	<b>6.23</b>	<b>3.74</b>
LATCH	0	1	1	0.11	0.02	0.00
<b>Affine-LATCH</b>	<b>37</b>	<b>14.82</b>	<b>219,6</b>	<b>4.52</b>	<b>2.16</b>	<b>0.17</b>
FREAK	0	1	1	0.34	0.15	0.18
<b>Affine-FREAK</b>	<b>145</b>	<b>7.06</b>	<b>49.8</b>	<b>4.37</b>	<b>2.38</b>	<b>1.94</b>

**Table 1.4** Matching methods performance over query and target images from Figure 1.16. The proposed IMAS methods proposed here appear in bold. Computations were performed on an Intel(R) Core(TM) i5-4210U CPU 1.70GHz with 2 cores.

M - Matches.

*ar* - area ratio.

	M	<i>ar</i>	$ar^2$	Keypoints (seconds)	Matching (seconds)	Filters (seconds)
SIFT	102	1	1	0.23	0.01	0.09
ASIFT	317	13.7	189.6	5.43	1.68	0.47
<b>(Optimal) Affine-SIFT</b>	<b>292</b>	<b>7.06</b>	<b>49.8</b>	<b>2.71</b>	<b>0.38</b>	<b>0.30</b>
RootSIFT	110	1	1	0.25	0.01	0.09
<b>Affine-RootSIFT</b>	<b>219</b>	<b>6.9</b>	<b>47.6</b>	<b>2.23</b>	<b>0.28</b>	<b>0.24</b>
SURF	110	1	1	0.24	0.03	0.14
<b>(Optimal) Affine-SURF</b>	<b>663</b>	<b>14.82</b>	<b>219,6</b>	<b>3.68</b>	<b>1.19</b>	<b>0.73</b>
BRISK	29	1	1	1.57	0.00	0.04
<b>Affine-BRISK</b>	<b>49</b>	<b>8.42</b>	<b>70,89</b>	<b>17.57</b>	<b>0.06</b>	<b>0.08</b>
BRIEF	0	1	1	0.03	0.00	0.00
<b>Affine-BRIEF</b>	<b>7</b>	<b>14.82</b>	<b>219,6</b>	<b>2.06</b>	<b>0.09</b>	<b>0.03</b>
ORB	102	1	1	0.02	0.01	0.8
<b>Affine-ORB</b>	<b>90</b>	<b>14.82</b>	<b>219,6</b>	<b>2.12</b>	<b>0.31</b>	<b>0.40</b>
AKAZE	20	1	1	0.16	0.00	0.03
<b>Affine-AKAZE</b>	<b>51</b>	<b>8.42</b>	<b>70,89</b>	<b>2.31</b>	<b>0.06</b>	<b>0.09</b>
LATCH	54	1	1	0.07	0.01	0.04
<b>Affine-LATCH</b>	<b>101</b>	<b>14.82</b>	<b>219,6</b>	<b>1.72</b>	<b>0.12</b>	<b>0.10</b>
FREAK	124	1	1	0.14	0.01	0.10
<b>Affine-FREAK</b>	<b>182</b>	<b>7.06</b>	<b>49.8</b>	<b>2.54</b>	<b>0.11</b>	<b>0.31</b>

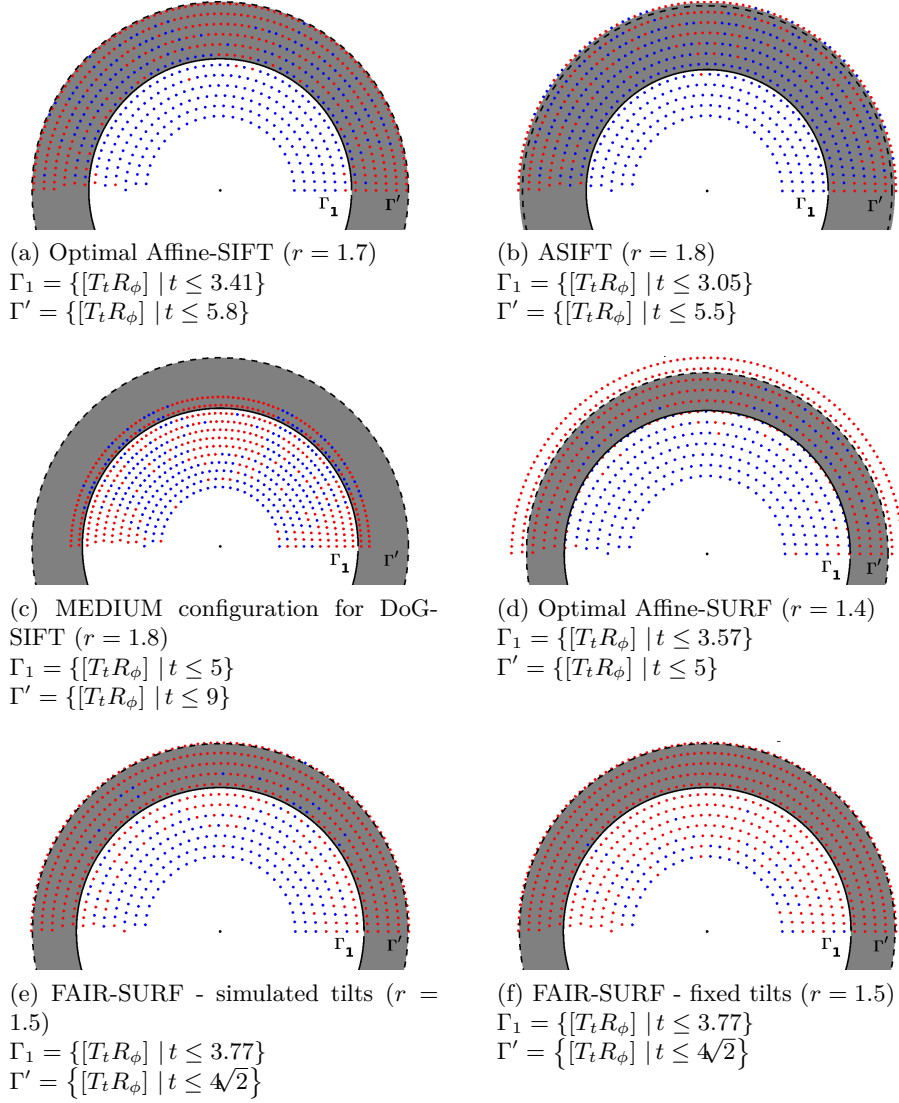
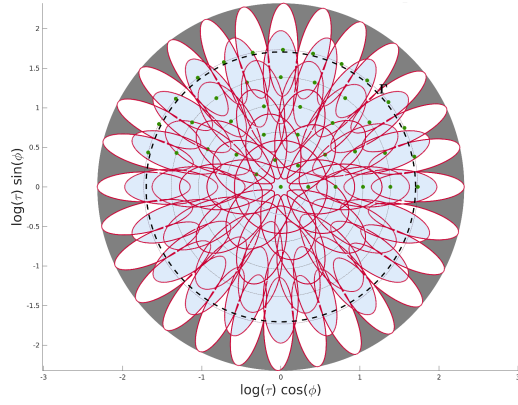


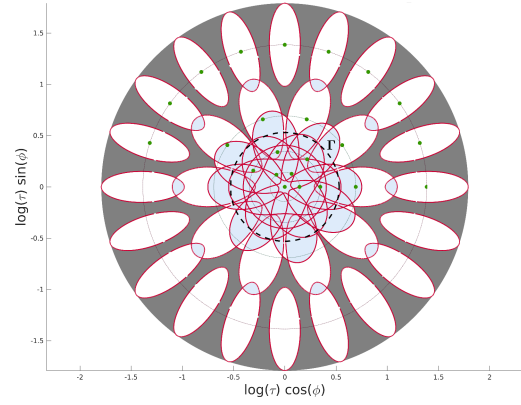
Figure 1.13: Extreme test results.

Black dot -  $[Id]$ .

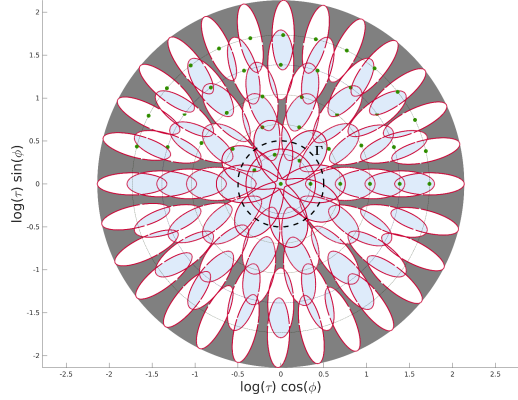
Coloured dots stand for  $[A_{t,\phi}^{-1}]$  and belong to a fixed  $\log 1.1$  uniform discretization of the annulus  $\{[T_t R_\phi] \mid 2 \leq t \leq 4\sqrt{2}\}$ . The angle  $\phi$  implicitly fixes  $[B_\phi^{-1}] = [T_\Lambda R_\phi]$  where  $\Lambda = \arg \max_t [T_t R_\phi] \in \Gamma'$ .  
Blue/Red dots - Success/Failure of ORSA Homography in identifying the underlying affine map.



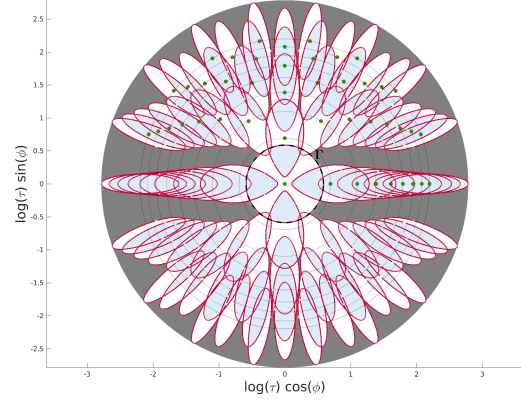
(a) Proposed covering for ASIFT in [MY09, YM11]. This is a log 1.8-covering of  $\{[T_t R_\phi] \mid t \leq 5.5\}$  with 41 affine simulations representing an area ratio of 13.77.



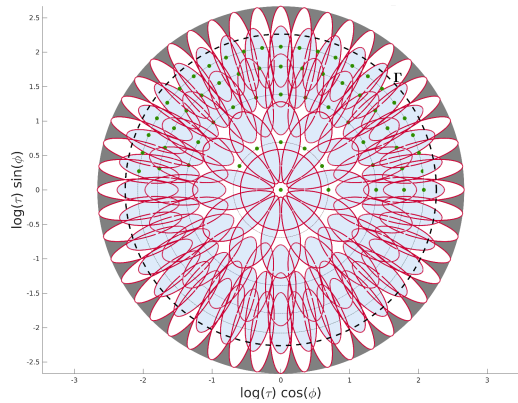
(b) Proposed covering for FAIR-SURF in [PLYP12], called fixed tilts. This is a log 1.5-covering of  $\{[T_t R_\phi] \mid t \leq 1.7\}$  with 23 affine simulations representing an area ratio of 11.42.



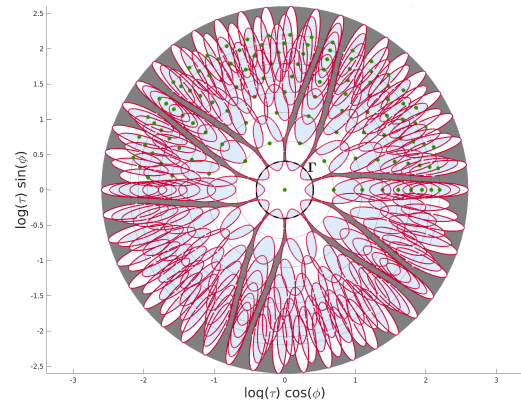
(c) Proposed covering for FAIR-SURF in [PLYP12], called simulated tilts. This is a log 1.5-covering of  $\{[T_t R_\phi] \mid t \leq 1.65\}$  with 41 affine simulations representing an area ratio of 13.77.



(d) Proposed covering in [MMP15], called MEDIUM configuration for DoG-SIFT. This is a log 1.8-covering of  $\{[T_t R_\phi] \mid t \leq 1.8\}$  with 45 affine simulations representing an area ratio of 9.



(e) Proposed covering in [MMP15], called HARD configuration for DoG-SIFT. This is a log 1.8-covering of  $\{[T_t R_\phi] \mid t \leq 9.6\}$  with 61 affine simulations representing an area ratio of 13.



(f) Proposed covering in [MMP15], called HARD Configuration for SURF-SURF. This is a log 1.5-covering of  $\{[T_t R_\phi] \mid t \leq 1.5\}$  with 112 affine simulations representing an area ratio of 21.28.

Figure 1.14: Examples of coverings found in the literature for maximum tilt tolerances as in Figure 1.11.

Gray areas - Uncovered.

Blue areas - Covered by at least two disks.

White areas - Covered by only one disk.



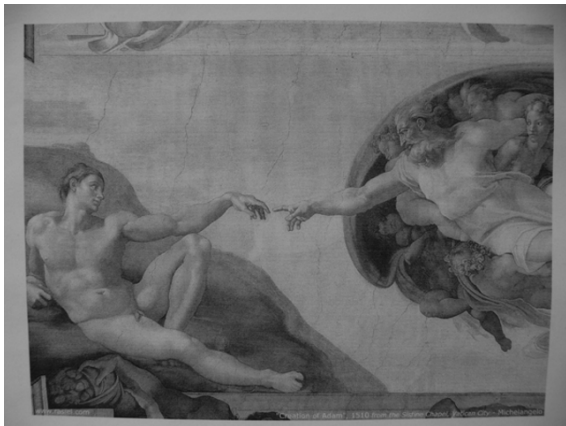


(a)  $800 \times 640$



(b)  $800 \times 640$

Figure 1.15: Graffiti. Both images generate a large number of keypoints for most methods.



(a)  $600 \times 450$



(b)  $600 \times 450$

Figure 1.16: Adam. Both images generate a small number of keypoints for most methods.

## 2 Fast affine invariant image matching

### 2.1 Introduction

We saw that the best established image comparison method is SIFT [Low04]. This method was shown in [MY08] to perform recognition invariant to image rotations, translations, and camera zoom-outs. SIFT has inspired numerous variations over the past 15 years [KS04, BTV06, AZ12]. As in Chapter 1, we refer to these methods as Scale Invariant Matching Methods (SIIM). Several attempts have also been made to create local image descriptors invariant to affine transformations [MCUP04, MSCG03, MTS<sup>+</sup>05]. Yet, it was shown in [MY08] that none of these approaches is truly invariant to local affine transformations, due to the fact that optical blur and affine transformations do not commute. As a result, these methods cannot handle angle viewpoint differences larger than  $60^\circ$  for planar objects [MY09, MMP15], and lose quickly efficiency for angles larger than  $45^\circ$  [Kar16].

A more pragmatic approach, proposed a few years ago with the ASIFT Algorithm [MY08] and adopted by several authors ever since [PLYP12, MMP15], consists in applying a pre-determined set of affine transformations to each compared image, in order to simulate the transformations induced by the viewpoint changes. Instead of comparing two images, the resulting algorithm therefore compares all the pairs of simulated images. As in Chapter 1, we refer to these simulation algorithms as IMAS, for Image Matching by Affine Simulation. In favorable cases, IMAS can capture changes of point of view up to an impressive  $88^\circ$ .

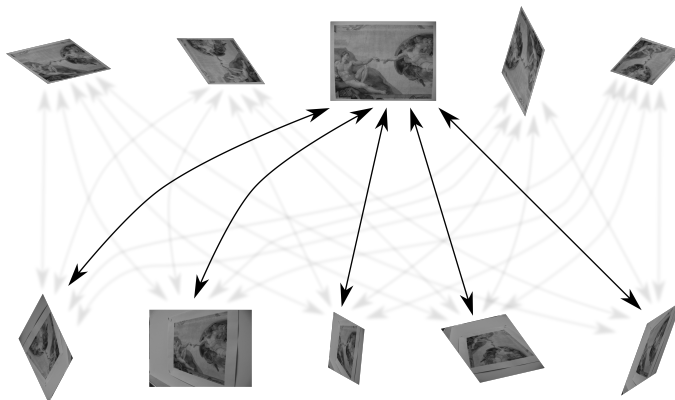


Figure 2.1: IMAS algorithms start by applying a finite set of optical affine simulations to  $u$  and  $v$ , followed by pairwise comparisons.

The first IMAS method provided with a mathematical proof of affine invariance is ASIFT [MY09, YM11]. As its name indicates, ASIFT is an affine invariant extension of SIFT, that actually operates on SIFT. It can operate on any Scale Invariant Image Matching (SIIM) method like SURF [BTV06] as well, provided its descriptor shows some robustness to angle viewpoint changes like SIFT descriptors do. Unlike MSER [MCUP04], LLD [MSCG03], Harris-Affine and Hessian-Affine [MTS<sup>+</sup>05], which attempt at normalizing all of the six affine parameters, ASIFT only simulates the two camera axis angles, and then applies SIFT which simulates the scale and normalizes the rotation and the translation. Similarly, FAIR-SURF [PLYP12] is an IMAS method replacing SIFT by SURF in ASIFT. MODS [MMP15] is another IMAS using heuristics to test fewer affine transforms and also combining different SIIMs. Other IMAS approaches do not involve local descriptors: FAsT-Match [KRTA13] delivers affine invariance by assuming that the template (a patch in the query image) can be recovered inside the target image by a *unique* affine map. The six affine parameters are simulated instead of the three involved in ASIFT.

Our goal is in this chapter is to provide a generic IMAS method that cumulates three new improvements, all aimed at acceleration and robustness:

- it minimizes the number of camera axis angles to be simulated;
- it defines local hyper-descriptors grouping descriptors at the same location to accelerate matching;
- it resolves the problem of Lowe’s matching thresholds that inhibits matching in presence of multiple similar objects in the same image.

In IMAS methods, the viewpoint change between different views of the same planar scene is measured by the so-called *relative transition tilts* [MY09, YM11]. Transition tilts to be simulated to match two images can be much larger than absolute tilts. We describe here our optimal solution to the key question of IMAS methods: how to choose the list of tilts applied to both images to test these large transition tilts before comparison? In Chapter 1 we treated this question by finding optimal coverings of the space of affine tilts. In Section 2.2 we recall these results and give implementation details to construct nearly optimal coverings. Section 2.3 gives the construction of hyper-descriptors. In Section 2.4 we describe the two structural and computational improvements of the method. One is the replacement of Lowe’s acceptance criterion by an *a contrario* criterion, and the other one is the elimination of useless “flat descriptors”. Section 2.5 contains a short experimental assessment.

## 2.2 Image Matching by Affine Simulation

In this section, we will recall and present in a more practical way some definitions and results formally introduced in Chapter 1. The notion of transition tilt appearing in Definition 1.3 is helpful for measuring the affine distortion from a fixed affine viewpoint to surrounding affine viewpoints. Transition tilts do not depend on the class representative of  $A$  or  $B$ , so they can be defined directly on the quotient  $GL^+(2) / \sim$ . Proposition 1.5 gives an adequate measuring tool. As argued in Chapter 1 and in [Kar16], transition tilt tolerances (determining visible viewpoints) are SIIM dependent. Most SIIMs are able to identify viewpoint changes under  $45^\circ$  for image sizes around  $700 \times 550$ .

Let us now recall disks formulas in the space of tilts with respect to the metric  $d$  in Proposition 1.5.

**Notation 2.1.** Let  $S \in \Omega$  and  $r > 0$ . We denote either by  $\mathcal{B}(S, r)$  or by  $\mathcal{B}_S^r$ , the disk in the space of tilts centered at  $S$  and with radius  $r$ .

**Theorem** (Theorem 1.1 from Chapter 1). Given an element of the space of tilts in canonical form  $[T_t R(\phi_1)]$ , the disk  $\mathcal{B}([T_t R(\phi_1)], r)$  in the space of tilts corresponds to the following set

$$\left\{ [T_s R(\phi_2)] \mid G(t, s, \phi_1, \phi_2) \leq \frac{e^{2r} + 1}{2e^r} \right\},$$

where

$$G(t, s, \phi_1, \phi_2) = \left( \frac{\frac{t}{s} + \frac{s}{t}}{2} \right) \cos^2(\phi_1 - \phi_2) + \left( \frac{\frac{1}{st} + st}{2} \right) \sin^2(\phi_1 - \phi_2).$$

Figures 1.4 and 1.3 from Chapter 1 show, in a perspective view and in polar coordinates respectively, four disks in the space of tilts centered at four reference tilts. The radius of these disks corresponds to a maximal change of angle view of  $45^\circ$  with respect to the disk's center. The larger the tilt, the smaller the disk with that radius, which means that we need more and more disks to cover the high tilt regions. Notice that all disks appear duplicated by symmetry. Indeed, a perspective visualization of  $\Omega$  is impossible in  $\mathbb{R}^3$ :  $\Omega$  is the quotient of the half sphere by a central symmetry.

As a consequence, affine simulation is now a reliable way of extending the initial visibility range of a SIIM. The idea is to place affine simulations in a way that they render all elements in region  $\gamma \subset \Omega$  perfectly visible for at least one of them. When that happens we call that set of affine simulations a *covering* of the region in question.

**Definition 2.1.** We call  $\mathbb{S} \subset \Omega$  an  $\alpha^\circ$ -covering of a region  $\Gamma \subset \Omega$  if and only if

$$\Gamma \subset \bigcup_{S \in \mathbb{S}} \mathcal{B}\left(S, \log \frac{1}{\cos(\alpha^\circ)}\right).$$

**Remark 2.1.** In Definition 2.1,  $\mathbb{S} \subset \Omega$  actually corresponds to the centers of the balls constituting the covering. For our scopes it will be a finite set, that determines the set of affine transformations used to simulate affine viewpoints in IMAS algorithms, i.e.

$$\{i(S) \mid S \in \mathbb{S}\}.$$

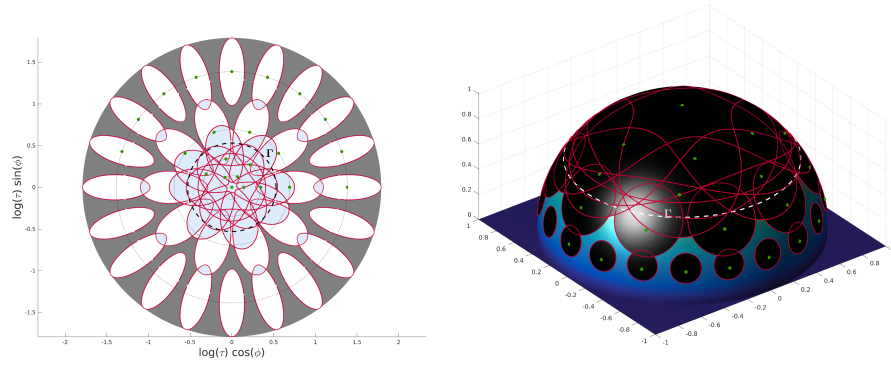
The region  $\Gamma \subset \Omega$  of Definition 2.1 usually denotes a circular region representing all viewpoints within a certain angle  $\theta$ . The following definition gives a name to these sets.

**Definition 2.2.** The set  $\Gamma \subset \Omega$  is called a  $\gamma^\circ$ -region if and only if

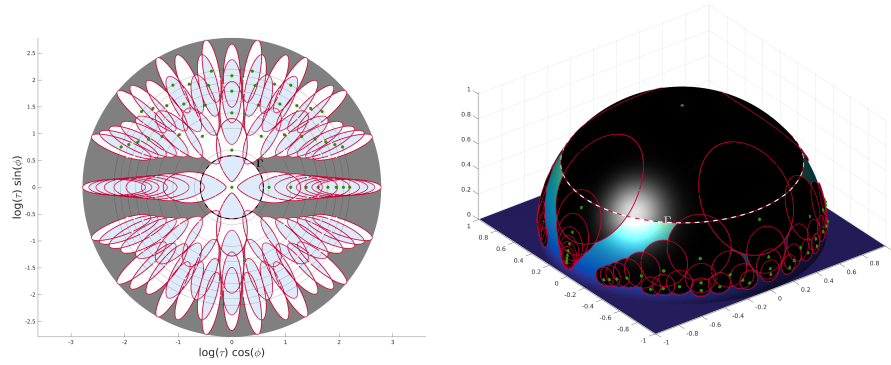
$$\Gamma = \left\{ [T_t R_\phi] \mid t \leq \frac{1}{\cos(\gamma^\circ)} \right\}.$$

Several  $\alpha^\circ$ -coverings of  $\gamma^\circ$ -regions have been proposed in [MY09, YM11, PLYP12, MMP15] and in Chapter 1 for SIFT and SURF; among them those in Figure 2.2. It is easily seen that they are far from optimality: some of these coverings do not really cover the region they were meant to, except for ASIFT [MY09, YM11] which instead is visually redundant. The following section describes the near optimal coverings proposed in Chapter 1.

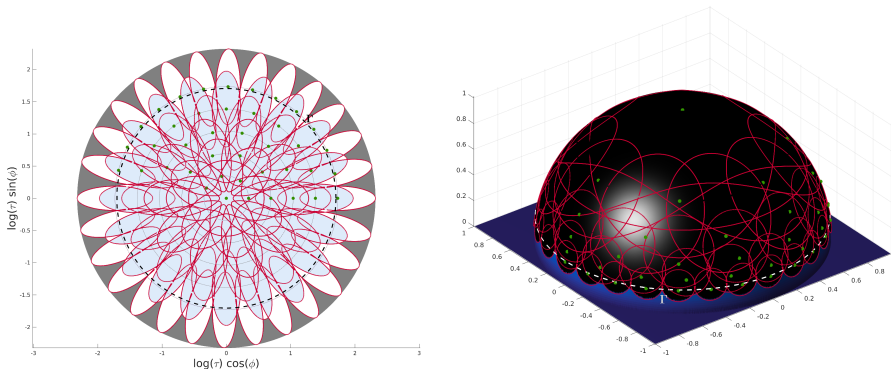




(a) FAIR-SURF fixed tilts [PLY12], a set of 23 affine simulations with an area ratio of 11.42. It would represent a  $48^\circ$ -covering of a  $53^\circ$ -region. Highly redundant in the central part, it does not cover the  $80^\circ$ -region.



(b) MEDIUM DoG-SIFT [MMP15], a set of 45 affine simulations with an area ratio of 9. It would represent a  $56^\circ$ -covering of a  $56^\circ$ -region. Although highly redundant, it does not cover the  $80^\circ$ -region.



(c) ASIFT [MY09, YM11], a set of 41 affine simulations with an area ratio of 13.77. It would represent a  $56^\circ$ -covering of a  $80^\circ$ -region. Highly redundant, it does cover the region it was meant to.

Figure 2.2: Non optimal coverings

Green points - Affine camera simulations  
 Red lines - Visibility tolerance from each affine simulation  
 White/Black surfaces - Visible viewpoints regions  
 Dashed line - Covered regions

### 2.2.1 Near Optimal $\alpha^\circ$ -Coverings

Near optimal coverings in Chapter 1 ensure minimal complexity for IMAS algorithms. One example of these  $\alpha^\circ$ -coverings is represented in Figure 2.5a.

The optimization problem is to minimize the overall number of descriptor comparisons while maintaining the same detection efficiency. This minimization *is not* equivalent to a minimization of the number of simulated versions being used. As stated in Chapter 1, the optimization problem should be driven by the area ratio from Definition 1.9. The area ratio allows to measure the total number of key points from all tilted versions, including the original one. Therefore, sets  $\mathbb{S}$  with small area ratios will involve less key point comparisons. Although this minimization problem is NP hard, the search space is drastically reduced to those  $\log r$ -coverings that can be obtained by feasible sets from Definition 1.10. Feasible sets are composed by groups of concentric equidistant tilts (a sound isotropy requirement), with the addition of the identity map, needed to avoid image resolution loss before comparison.

Some conditions have been proposed in Chapter 1 in order to verify that a  $\gamma^\circ$  region is truly covered by a feasible set with a fixed number of concentric equidistant tilts. Let the intersection of disks boundaries, which are composed at most of two elements for non identical disks, be denoted by

$$\Sigma_i := \partial \mathcal{B}_{[T_{t_i}]}^{\log r} \cap \partial \mathcal{B}_{[T_{t_i} R_{\phi_i}]}^{\log r}, \quad (2.1)$$

and their respective closest and farthest elements be denoted by

$$\min \Sigma_i := \arg \min_{S \in \Sigma_i} d(S, [Id]), \quad \max \Sigma_i := \arg \max_{S \in \Sigma_i} d(S, [Id]).$$

Then Algorithm 2 summarizes the aforementioned conditions in a function, called IS-GAMMACOVERED, that is to be called for querying if a feasible set covers a  $\gamma^\circ$  region.

Figure 2.3 illustrates the iterative process described in Algorithm 2. One crucial step in Algorithm 2 is the creation of an  $\varepsilon$ -dense set of a given annulus. As explained in Chapter 1, this set is a helping hand to ensure a  $\log(r - \varepsilon)$ -covering up to an error of  $\varepsilon$  and so, by dilating back disks radius to  $r$  one ensures  $\log r$ -coverings. Of course, there exists an infinite number of  $\varepsilon$ -dense sets. For annulus like

$$\mathcal{B}_{[Id]}^{\log t_{i+1}} \setminus \mathcal{B}_{[Id]}^{\log t_i}$$

we propose to build the following set, which is proven to be a  $\varepsilon$ -dense set by the mere application of the triangle inequality of the metric  $d$ ,

$$\mathbb{F}_\varepsilon := \{[T_{e^{n\varepsilon} t_i} R_{k\beta_i}] \in \Omega \mid n, k \in \mathbb{N}_+, e^{n\varepsilon} t_i < t_{i+1}, k\beta_\varepsilon(e^{n\varepsilon} t_i) < \pi\}, \quad (2.2)$$

where the function  $\beta_\varepsilon$ , appearing in Definition 2.3, determines the angle step for equal distances over the same tilt.

Indeed, let  $z \in \mathcal{B}_{[Id]}^{\log t_{i+1}} \setminus \mathcal{B}_{[Id]}^{\log t_i}$  and its surrounding four points  $y_i \in \mathbb{F}_\varepsilon$ ,  $i \in \mathbb{Z}/4\mathbb{Z}$  satisfying  $d(y_i, y_{i+1}) = \varepsilon$ . Four auxiliary points,  $x_i$   $i \in \mathbb{Z}/4\mathbb{Z}$ , are defined as projections of  $z$  on arcs (see Figure 2.4). In that case, we always have either  $d(y_i, x_i) \leq \frac{\varepsilon}{2}$ , either  $d(y_{i+1}, x_i) \leq \frac{\varepsilon}{2}$ . This implies that there exists at least one pair  $(j, k)$  with  $k = j$  or  $k = j + 1$  for which  $d(x_j, z) \leq \frac{\varepsilon}{2}$  and such that

$$d(y_k, z) \leq d(y_k, x_j) + d(x_j, z) \leq \varepsilon.$$

---

**Algorithm 2** ISGAMMACOVERED

---

**input:**

The initial  $\alpha^\circ$  visibility (fixes  $r = \frac{1}{\cos \alpha^\circ}$ ).

The  $\gamma^\circ$  region to cover (fixes  $t_\gamma = \frac{1}{\cos \gamma^\circ}$ ).

**parameters:**

$n$  - Number of concentric equidistant tilts for the feasible set (as in Definition 1.10). This also fixes the amount of sets  $\Sigma_i$ , defined in (2.1).

$\varepsilon$  - Number of uniformly discretized elements in each dimension with respect to the metric  $d$  of Proposition 1.5.

**start:**

covered\_portion =  $r$

▷ A feasible set always has the disk  $\mathcal{B}_{[Id]}^{\log r}$

**if**  $\Sigma_1 = \emptyset$  **then**

└ **return**(false)

**foreach**  $i = 1, \dots, n-1$  **do**

┌ **if**  $\tau(\min \Sigma_i) > \text{covered\_portion}$  **then**

└ **return**(false)

covered\_portion =  $\max \Sigma_i$

▷ the annulus  $\mathcal{B}_{[Id]}^{\log \tau(\max \Sigma_i)} \setminus \mathcal{B}_{[Id]}^{\log \tau(\min \Sigma_i)}$  is already covered

**if** covered\_portion  $> t_\gamma$  **then**

└ **return**(true)

**if**  $\Sigma_{i+1} = \emptyset$  **then**

└ **return**(false)

**foreach**  $[T_t R_\phi] \in \mathbb{F}_\varepsilon$  **do**

┌ **if**  $[T_t R_\phi] \notin \bigcup_{j=i}^{i+1} \mathcal{B}^{\log r - \varepsilon} \left[ T_{t_j} R \left[ \left\lfloor \frac{\phi}{\phi_j} \right\rfloor \phi_j \right] \right] \cup \mathcal{B}^{\log r - \varepsilon} \left[ T_{t_j} R \left[ \left\lceil \frac{\phi}{\phi_j} \right\rceil \phi_j \right] \right]$  **then**

▷  $\mathbb{F}_\varepsilon$  is the finite  $\varepsilon$ -dense set appearing in (2.2).

└ **return**(false)    ▷  $[T_t R_\phi]$  must lie inside one of the four nearest disks (Theorem 1.1 is used)

▷ at this point the annulus  $\mathcal{B}_{[Id]}^{\log \tau(\min \Sigma_{i+1})} \setminus \mathcal{B}_{[Id]}^{\log \tau(\max \Sigma_i)}$  has been proved to be covered

covered\_portion =  $\min \Sigma_{i+1}$

**if**  $\tau(\max \Sigma_n) > t_\gamma$  **then**

└ **return**(true)

**else**

└ **return**(false)

---

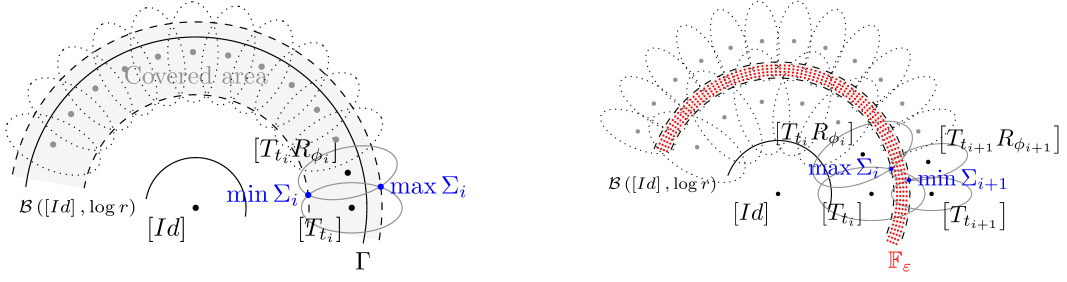


Figure 2.3: Verifying covering conditions for feasible sets in Algorithm 2. Left : covered annulus determined by  $\min \Sigma_i$  and  $\max \Sigma_i$ . Right : all elements in  $\mathbb{F}_\varepsilon$  must lie inside at least one disk.

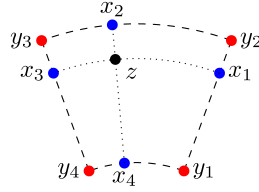


Figure 2.4: Proving the  $\varepsilon$ -density of  $\mathbb{F}_\varepsilon$ .

**Definition 2.3.** We denote by  $\beta_\varepsilon$ , the function defined by

$$\beta_\varepsilon: \begin{cases} ]1, \infty] & \rightarrow [0, \pi[ \\ t & \mapsto \phi(t) \end{cases},$$

where  $\phi(t)$  is such that

$$d([T_t], [T_t R_{\phi(t)}]) = \varepsilon.$$

The procedure in the proof of Proposition 1.8 in Chapter 1 can also be applied to find all kinds of near optimal coverings depending on the initial visibility  $\alpha^\circ$  and the  $\gamma^\circ$ -region to be covered. Algorithm 3 delivers a way of finding near optimal  $\alpha^\circ$ -coverings by fixing  $n$ , the number of concentric equidistant tilts, and then optimizing over  $2n$  dimensions. By means of the canonical injection in Definition 1.5, a given  $\alpha^\circ$ -covering determines the set of affine maps to be simulated in Algorithm 5. Some examples of these kind of coverings can be found in Table 2.1.

**Remark 2.2.** The global optimization proposed in Algorithm 3 should be followed by some iterations of a refined search on the neighboring subsets containing the current optimum.

### 2.2.2 Simulating Digital Tilts

The former sections only dealt with affine geometry and now we shall focus on digital image recognition. The formalism of the camera blur, presented in Section 1.3 from Chapter 1, states that there are actually three different notions of tilt. Digital tilts are the ones used in practice. It all adds up because the simulated tilt yields a blur permitting  $\mathbf{S}_1$ -sampling.

---

**Algorithm 3** Finding near optimal coverings
 

---

**input:**

 The initial  $\alpha^\circ$  visibility and the  $\gamma^\circ$  region to cover.

**parameters:**
 $n$  - Number of concentric equidistant tilts (as in Definition 1.10).

 $\kappa$  - Number of uniformly discretized elements in each dimension with respect to the metric  $d$  of Proposition 1.5.

**inner definitions:**
 $]t_1, t_2]_\kappa = \{ \frac{t_2}{e^{n\varepsilon_\kappa}} \mid n \in \mathbb{N}, t_1 < \frac{t_2}{e^{n\varepsilon_\kappa}} \leq t_2 \}, ]0, \beta]_\kappa = \{ n\beta_{\varepsilon_\kappa}(t) \mid n \in \mathbb{N}, 0 < n\beta_{\varepsilon_\kappa}(t) \leq \beta \}$ 

 where  $\varepsilon_\kappa$  is such that  $]t_1, t_2]_\kappa = ]0, \beta]_\kappa^t = \kappa$ , and  $\beta_{r^2}(t_i)$  appears in Definition 2.3.

**start:**
 $r = \log \left( \frac{1}{\cos(\alpha^\circ)} \right)$ 
 $\triangleright \alpha^\circ$  equivalent transition tilt

 $ar = \infty$ 
 $\triangleright$  current minimal area ratio

**foreach**  $(t_1, t_2, \dots, t_n) \in ]r, r^2]_\kappa \times ]t_1 r, t_1 r^2]_\kappa \times \dots \times ]t_{n-1} r, t_{n-1} r^2]_\kappa$  **do**  
     **foreach**  $(\phi_1, \phi_2, \dots, \phi_n) \in ]0, \beta_{r^2}(t_1)]_\kappa^{t_1} \times ]0, \beta_{r^2}(t_2)]_\kappa^{t_2} \times \dots \times ]0, \beta_{r^2}(t_n)]_\kappa^{t_n}$  **do**  
         **if**  $(\sum_{i=1}^n t_i \lfloor \frac{\pi}{\phi_i} \rfloor < ar)$  &  $IS\Gamma\Gamma\Gamma\Gamma\Gamma COVERED(\gamma, t_1, \dots, t_n, \phi_1, \dots, \phi_n)$  **then**  
              $ar = \sum_{i=1}^n t_i \lfloor \frac{\pi}{\phi_i} \rfloor$   
              $(t_1^{opt}, \dots, t_n^{opt}) = (t_1, \dots, t_n)$   
              $(\phi_1^{opt}, \dots, \phi_n^{opt}) = (\phi_1, \dots, \phi_n)$ 
**return**  $\mathcal{S} = \{t_k^{opt}, \phi_k^{opt}\}_{k=1, \dots, n}$ 


---

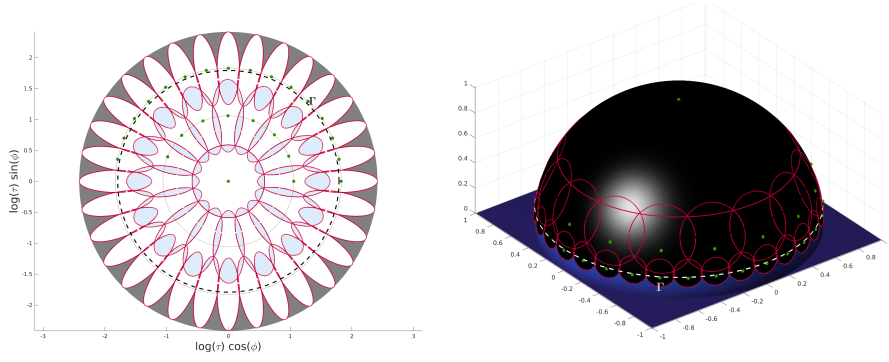
**Table 2.1** Near optimal  $\alpha^\circ$ -coverings of  $\gamma^\circ$ -regions. As proved in Chapter 1, these  $\alpha^\circ$ -coverings will ensure for a generic IMAS an extended visibility in column **EV** for a near-minimal area ratio in column **AR**.

$\alpha^\circ$	$\gamma^\circ$	<b>EV</b>	<b>AR</b>	$t_1^{opt}$	$\phi_1^{opt}$	$t_2^{opt}$	$\phi_2^{opt}$	$t_3^{opt}$	$\phi_3^{opt}$
45°	80°	87°	15.889	1.84641	0.459445	2.68973	0.234551	4.58177	0.116774
54°	80°	87°	7.354	2.54902	0.450362	4.71215	0.18624	-	-
54°	81°	88°	7.548	2.67673	0.350162	5.65043	0.175859	-	-
56°	80°	87°	6.290	2.89419	0.396183	6.33474	0.198091	-	-
56°	83°	88°	7.221	2.89419	0.397562	6.07477	0.150497	-	-
56°	84°	89°	9.014	2.79309	0.461217	4.61946	0.24717	9.65081	0.123523
58°	82°	88°	5.971	3.01682	0.450814	6.03598	0.200202	-	-
58°	84°	89°	7.979	3.02483	0.448874	5.09033	0.261983	10.4035	0.131014
60°	84°	89°	6.126	3.2948	0.396543	7.78261	0.156965	-	-

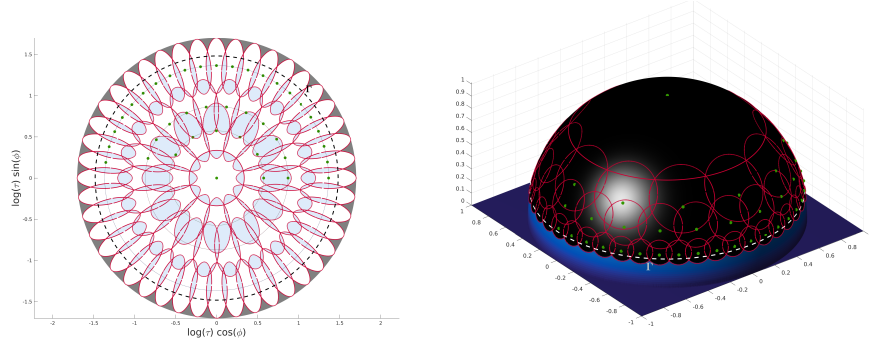
For the sake of clearness in this chapter, we denote by  $\mathbb{T}_t$  the operator  $\mathbb{T}_t^x$  appearing in Definition 1.8. Therefore, digital tilts involving blur in the  $x$  direction and transforming a digital image into another digital image, can be rewritten as

$$\mathbf{u} \rightarrow \mathbf{S}_1 \mathbb{T}_t I \mathbf{u}.$$

Our goal is to define rigorously affine invariant recognition for digital images. With that in mind, we set once and for all how affine simulations are to be computed. Algorithm 4 digitally simulates tilts in any direction, i.e simulates affine viewpoints from any place of the sphere.



(a)  $56^\circ$ -covering with a set of 28 affine simulations that renders viewpoints visible in a  $82^\circ$ -region.



(b)  $45^\circ$ -covering with a set of 43 affine simulations that renders viewpoints visible in an  $82^\circ$ -region.

Figure 2.5: Near optimal coverings as in Chapter 1

Green points - Affine camera simulations  
 Red lines - Visibility tolerance from each affine simulation  
 White/Black surfaces - Visible viewpoints regions  
 Dashed line - Covered regions

---

**Algorithm 4** Generating digital tilts in any direction ( $\mathbf{u} \rightsquigarrow \mathbf{S}_1 \mathbb{T}_t R_\phi I \mathbf{u}$ )

---

**input:** the digital image  $\mathbf{u}$  and parameters  $(t, \phi)$

**start:**

$\mathbf{u} = \text{ROTATE}(\mathbf{u}, \phi)$   $\triangleright$  with bilinear interpolation

$\mathbf{u} = \text{GAUSSIANBLUR1D}(\mathbf{u}, \sigma = 0.8\sqrt{t^2 - 1})$   $\triangleright$  ROTATE will frame the rotated version of  $\mathbf{u}$  in a minimal rectangular image

$\mathbf{u} = \text{SUBSAMPLE}(\mathbf{u}, t)$   $\triangleright$  Gaussian blur in the  $x$ -direction

**return**  $\mathbf{u}$   $\triangleright$  Subsamples the image along the  $x$ -direction by a factor of  $t$

---

### 2.2.3 From SIIM to IMAS

We have seen in the previous sections how to compute a near optimal discrete set of affine transformations in order to cover a given region with a visibility  $\alpha$ . The core idea of the IMAS approach, described in Algorithm 5 and illustrated by Figure 2.1, is to apply this optimal set of transformations to images before comparing them with a SIIM method. We showed in Proposition 1.7 of Chapter 1 that this algorithm offers an affine-invariant version of the associated SIIM method. Indeed, the optimal covering ensures that there is at least one pair of simulated images whose transition tilt is visible for the SIIM. Table 2.1 gives some examples of ensured visibility, depending on the assumed initial visibility  $\alpha^\circ$  of the SIIM and the  $\gamma^\circ$ -region to be covered.

Choosing the right  $\alpha$ -covering is fundamental. It depends on the SIIM's transition tilt

---

**Algorithm 5** Formal IMAS (Image Matching by Affine Simulation)

---

**input:** query and target images:  $\mathbf{u}$  and  $\mathbf{v}$ .

**parameters:**

a routine SIIM-DETECTOR, two sets of optimal coverings  $\mathcal{S}_1 = \{t_k^1, \phi_k^1\}_{k=1, \dots, n_1}$  and  $\mathcal{S}_2 = \{t_k^2, \phi_k^2\}_{k=1, \dots, n_2}$ , provided by Algorithm 3

$\triangleright \zeta(D, \mathbf{u}, t, \phi)$  is a simple routine that filters out those descriptors in  $D$  which after back projection with  $(T_t R_\phi)^{-1}$  are not fully inside the domain of  $\mathbf{u}$

**start:**

**foreach**  $k = 1, \dots, n_1$  **do**

$\left[ \begin{array}{ll} D_k^u = \text{SIIM-DETECTOR}(\mathbf{S}_1 \mathbb{T}_{t_k^1} R_{\phi_k^1} \mathbf{u}) & \triangleright \text{Descriptors on each simulated image from } \mathbf{u} \\ D_k^u = \zeta(D_k^u, \mathbf{u}, t_k^1, \phi_k^1) \end{array} \right.$

**foreach**  $k = 1, \dots, n_2$  **do**

$\left[ \begin{array}{ll} D_k^v = \text{SIIM-DETECTOR}(\mathbf{S}_1 \mathbb{T}_{t_k^2} R_{\phi_k^2} \mathbf{v}) & \triangleright \text{Descriptors on each simulated image from } \mathbf{v} \\ D_k^v = \zeta(D_k^v, \mathbf{v}, t_k^2, \phi_k^2) \end{array} \right.$

**foreach**  $(k_1, k_2) \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}$  **do**

$\left[ M_{k_1, k_2} = \text{SIIM-MATCHER}(D_{k_1}^u, D_{k_2}^v) \right. \quad \triangleright \text{Set of matches between } \mathbf{u} \text{ and } \mathbf{v}$

**return**  $M = \bigcup_{(k_1, k_2) \in \{1, \dots, n_1\} \times \{1, \dots, n_2\}} M_{k_1, k_2}$ 

---

tolerance (with respect to a given image size) and the wanted extended visibility. Clearly, the parameter  $\alpha$  should be less than the initial visibility (determined by the transition tilt tolerance) of the SIIM.

## 2.3 Hyper-Descriptors in IMAS

IMAS algorithms implementations have to deal with various types of spurious or redundant matches that do not appear in the corresponding SIIM approaches. In this section, we describe the reasons behind the occurrence of these aberrant matches. Since false matches make it hard in practice to identify the underlying image transformation, we propose a simple idea to eliminate most of them without significantly reducing the number of good matches. To do this, we rely on the notion of hyper-descriptors. A hyper-descriptor is a group of several local descriptors of the different simulated images whose corresponding keypoints are close when back-projected in the original image plane.

### 2.3.1 Identifying Aberrant Matches

We identify in the following paragraphs three kinds of spurious or aberrant matches inherently produced by IMAS algorithms.

First, these approaches naturally favor repetitive matches, since the same keypoints can be matched in several pairs of simulated images. These matches are often good but provide redundant information and should be considered as a single match.

Second, IMAS algorithms are prone to yield what we call *multiple-to-one* matches, where several keypoints from the query image match one and the same keypoint from the target image. These matches appear frequently in IMAS algorithms when they use Lowe's second nearest neighbor acceptance criterion<sup>1</sup>. Since simulated target images with

---

<sup>1</sup>In Lowe's acceptance criterion, the ratio between the distance to the nearest neighbor and the distance to the second nearest neighbor is thresholded to decide if a match is accepted or rejected.



high absolute tilts are smaller and contain fewer keypoints, the number of second nearest neighbors they provide is also smaller for these images than for lower absolute tilts. In consequence, keypoints in these images are more likely to match those in the query image and have a tendency to be involved in *multiple-to-one* matches. These matches can be considered as false matches.

Third, IMAS algorithms have a strong tendency to give *one-to-multiple* matches, where a single keypoint from the query image matches multiple keypoints from the target image. Such matches, which can also be taken as indications of false matches, are naturally eliminated from SIIM approaches by Lowe’s acceptance criterion. The multiplicity of keypoints comparisons for a given structure in IMAS algorithms favors the apparition of such *one-to-multiple* matches.

In order to handle all the above spurious matches generated by Algorithm 5, a post-processing routine was proposed in [MY09, YM11] and adopted by all subsequent IMAS approaches. This post-processing is usually composed of three successive filters designed to remove repetitive matches, *multiple-to-one* matches and *one-to-multiple* matches. Such a post-processing is described in Algorithm 6 and can be applied after Algorithm 5 to identify the underlying transformation between the two images. Unfortunately, many good matches also get eliminated by this hack. For example, if only one false match comes to meet one end of a true match, then both matches get eliminated. Conversely, truly repetitive objects create various *multiple-to-one* / *one-to-multiple* matches that will always get discarded by this post-processing, and should not.

To avoid the loss of such correct matches, while eliminating the spurious and aberrant matches described above, we introduce in the next section the notion of hyper-descriptor.

### 2.3.2 Hyper-Descriptors Matching

**Definition 2.4.** Let  $\mathbf{u}$  be an image and  $\{\mathbf{u}_1, \dots, \mathbf{u}_n\}$  the optimal set of affine transformed versions of  $\mathbf{u}$  obtained with an IMAS algorithm. We call  $\rho$ -hyper-descriptor of  $\mathbf{u}$  a group of SIIM descriptors of this set of images, whose keypoints, once reprojected in  $\mathbf{u}$ , are all contained in a ball of radius  $\rho$ . The corresponding group of keypoints is called  $\rho$ -hyper-keypoint.

In practice, we choose the radius  $\rho$  between 3 and 6 pixels, and we keep this parameter fixed. In the following, for the sake of simplicity, we will assume that  $\rho$  is given and speak directly about hyper-descriptors. Figure 2.6 shows an exemple of hyper-descriptor composed of three SIIM descriptors extracted from three different affine simulations of the same image  $\mathbf{u}$ . This example illustrates an ideal case in which the center of three SIIM descriptors coincide. In practice the keypoints are not detected at exactly the same location. This ideal case would appear if no numerical errors were involved when simulating tilts and obtaining descriptors around an infinitely accurate corresponding keypoint.

Now that we have defined hyper-descriptors, Algorithm 8 describes a greedy approach to extract them from an image  $\mathbf{u}$ . First, a SIIM detector (SIFT or SURF for instance) is applied to all affine simulated versions of  $\mathbf{u}$  (the set of affine transformations is provided as a parameter of the algorithm). Then, each detected SIIM descriptor gets assigned to an existing hyper-descriptor or a new one is created. When assigned to an existing group, the center of the hyper-keypoint needs to be recomputed and the hyper-descriptor can be merged with a neighboring group. Each of the above computations can be done with an  $O(1)$  complexity, implying that the descriptor extraction parts of Algorithm 5 and Algorithm 7 have about the same complexity.



---

**Algorithm 6** Post-Processing of Algorithm 5

---

**input:** $M$  - List of output Matches of Algorithm 5.**parameters:** $\rho_1, \rho_2, \rho_3$  - Distance thresholdsGEOMETRICFILTER - An algorithm detecting a geometric consensus among a list of matches (e.g. RANSAC, ORSA Homography [MMM12], ORSA Fundamental [MMM16] or USAC [RCP<sup>+</sup>13]).**output:**

Filtered list of Matches

 $\triangleright$  a match  $m$  is composed by  $m.k_q$  and  $m.k_t$  which are respectively the associated query key-point and target key-point. $\triangleright$  The spatial distance between any two key-points is denoted by  $\Lambda(k_1, k_2) = \left| \begin{pmatrix} k_1.x - k_2.x \\ k_1.y - k_2.y \end{pmatrix} \right|$  $\triangleright$  unique filter $M_u = \emptyset$ **start:****foreach**  $m \in M$  **do**

flag-unique = true

**foreach**  $m_u \in M_u$  **do**        **if**  $\Lambda(m.k_q, m_u.k_q) \leq \rho_1$  **and**  $\Lambda(m.k_t, m_u.k_t) \leq \rho_1$  **then**  
            flag-unique = false    **if** flag-unique == true **then**         $M_u = M_u \cup m$  $M = M_u$  $\triangleright$  multiple2one filter $M_m = \emptyset$ **foreach**  $m \in M$  **do**

flag-multiple2one = false

**foreach**  $m_m \in M \setminus \{m\}$  **do**        **if**  $\Lambda(m.k_q, m_m.k_q) \geq \rho_3$  **and**  $\Lambda(m.k_t, m_m.k_t) \leq \rho_2$  **then**  
            flag-multiple2one = true    **if** flag-multiple2one == false **then**         $M_m = M_m \cup m$  $M = M_m$  $\triangleright$  one2multiple filter $M_m = \emptyset$ **foreach**  $m \in M$  **do**

flag-one2multiple = false

**foreach**  $m_m \in M \setminus \{m\}$  **do**        **if**  $\Lambda(m.k_q, m_m.k_q) \leq \rho_2$  **and**  $\Lambda(m.k_t, m_m.k_t) \geq \rho_3$  **then**  
            flag-one2multiple = true    **if** flag-one2multiple == false **then**         $M_m = M_m \cup m$  $M = M_m$  $\triangleright$  Filtering matches agreeing with geometric consensus $M = \text{GEOMETRICFILTER}(M)$ **return**  $M$ 

---

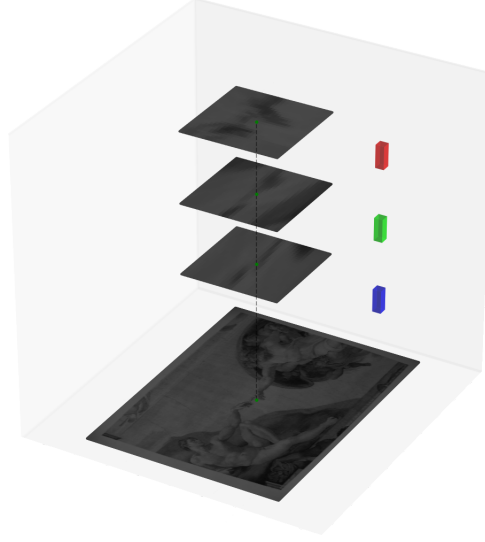


Figure 2.6: A  $\rho$ -hyper-descriptor  $d = \{\omega_1, \omega_2, \omega_3\}$  from the ground image  $\mathbf{u}$ . Green points denote keypoints associated to  $d$  and to each  $\omega_i$ .  $\omega_1$  (red square cuboid) describes the top patch, obtained from  $\mathbf{S}_1 \mathbb{T}_2 \mathbf{u}$ ;  $\omega_2$  (green square cuboid) describes the middle patch, obtained from  $\mathbf{S}_1 \mathbb{T}_4 R_{\frac{\pi}{4}} \mathbf{u}$ ;  $\omega_3$  (blue square cuboid) describes the bottom patch, obtained from  $\mathbf{S}_1 \mathbb{T}_6 R_{\frac{3\pi}{2}} \mathbf{u}$ .

---

**Algorithm 7** IMAS (with hyper-descriptors)

---

**input:** query and target images:  $\mathbf{u}$  and  $\mathbf{v}$ .

**start:**

$D_1 = \text{IMAS-DETECTOR}(\mathbf{u})$

▷ as in Algorithm 8

$D_2 = \text{IMAS-DETECTOR}(\mathbf{v})$

▷ as in Algorithm 8

$M = \text{IMAS-MATCHER}(D_1, D_2)$

▷ as in Algorithm 9

**return**  $M$

---

The distance between hyper-descriptors is defined as the minimal distance between the descriptors they are composed of.

**Definition 2.5.** *Let*

$$\begin{aligned} d_1 &= \{\alpha_1, \dots, \alpha_{n_1}\} \\ d_2 &= \{\beta_1, \dots, \beta_{n_2}\} \end{aligned}$$

*be two hyper-descriptors where  $\alpha_i$  and  $\beta_j$  are denoting SIIM descriptors. Let also  $\delta$  be a distance for SIIM descriptors. We call distance between  $d_1$  and  $d_2$  the positive number*

$$\Delta(d_1, d_2) = \min_{\substack{1 \leq i \leq n_1 \\ 1 \leq j \leq n_2}} \delta(\alpha_i, \beta_j).$$

**Remark 2.3.** *Usually  $\delta(\alpha, \beta)$  is either  $\|\alpha - \beta\|_{L_1}$ ,  $\|\alpha - \beta\|_{L_2}$  or the Hamming distance<sup>2</sup>.*

We can now derive an IMAS Matcher algorithm between hyper-descriptors (see Algorithm 9). We use here a straightforward generalization of Lowe's criteria to the previous distance. If two hyper-descriptors  $(d_1, d_2)$  define a match, then the left and right positions of a match are refined to

$$\arg \min_{(\alpha, \beta) \in d_1 \times d_2} \delta(\alpha, \beta).$$

---

<sup>2</sup>The Hamming distance is mostly used with binary descriptors.

---

**Algorithm 8** IMAS-Detector

---

**input:** image  $\mathbf{u}$ **parameters:**a routine SIIM-DETECTOR, one set of optimal coverings  $\mathcal{S} = \{t_k, \phi_k\}_{k=1,\dots,n}$ , provided by Algorithm 3 $\triangleright$  The spatial distance between any two descriptors is denoted by  $\Lambda(d_1, d_2) = \left| \begin{pmatrix} d_1.x - d_2.x \\ d_1.y - d_2.y \end{pmatrix} \right|$ **start:** $D = \emptyset$  $\triangleright$  Storage of hyper-descriptors**foreach**  $k = 1, \dots, n$  **do****foreach**  $s \in \text{SIIM-DETECTOR}(\mathbf{S}_1 \mathbb{T}_{t_k} R_{\phi_k} \mathbf{u})$  **do****if**  $(\mathbb{T}_{t_k} R_{\phi_k})^{-1}(s)$  is fully inside the domain of  $\mathbf{u}$  **then****if** it exists  $d \in \arg \min_{b \in D} \Lambda(b, s)$  such that  $\Lambda(d, s) \leq \rho$  **then** $d = d \cup \{s\}$  $\triangleright$  Add the descriptor  $s$  to the  $\rho$ -hyper-descriptor  $d$  $d.x = \frac{\sum_{s \in d} s.x}{|d|}, d.y = \frac{\sum_{s \in d} s.y}{|d|}$ **foreach**  $d_1 \in D$  such that  $\Lambda(d_1, d) \leq \rho$  **do** $d = d \cup d_1$  $d.x = \frac{\sum_{s \in d} s.x}{|d|}, d.y = \frac{\sum_{s \in d} s.y}{|d|}$ **else** $D = D \cup \{\{s\}\}$  $\triangleright$  Create a new  $\rho$ -hyper-descriptor from  $s$ **return**  $D$ 

---

---

**Algorithm 9** IMAS-Matcher

---

**input:** two sets of hyper-descriptors  $D_1, D_2$ .**parameters:** the match ratio  $\lambda \in ]0, 1[$ .**start:** $M = \emptyset$  $\triangleright$  Storage of Matches**foreach**  $d \in D_1$  **do** $a \in \arg \min_{c \in D_2} \Delta(d, c)$  $\triangleright \Delta(x, y) = \min_{(\alpha, \beta) \in x \times y} \delta(\alpha, \beta)$  $b \in \arg \min_{c \in D_2 \setminus a} \Delta(d, c)$ **if**  $\frac{\Delta(d, a)}{\Delta(d, b)} \leq \lambda$  **then** $M = M \cup \arg \min_{(\zeta, \eta) \in d \times a} \delta(\zeta, \eta)$ **return**  $M$ 

---

In order to get similar performances, the parameter *match ratio* in Algorithm 9 should be greater than its homologous in the SIIM-Matcher of Algorithm 5. Indeed, the second nearest neighbour applied on each simulated target image is less restrictive than just one application of it on all simulated target images at the same time.

In practice, for SIFT based descriptors, the match ratio ( $\lambda$ ) of Algorithm 9 is set to 0.8. This configuration seems to correspond to a match ratio of 0.6 for the SIIM-Matcher of Algorithm 5.

Note that hyper-descriptors are not associated with a given affine transformation but rather group descriptors from several simulated versions of  $\mathbf{u}$ . Comparing all the hyper-descriptors of two images  $\mathbf{u}$  and  $\mathbf{v}$  is therefore faster than comparing the descriptors of all their simulated versions. Indeed, when computing the distance between an hyper-descriptor of  $\mathbf{u}$  and an hyper-descriptor of  $\mathbf{v}$ , the computation can be stopped as soon as this distance exceeds the second smallest distance already calculated for this point. This

step saves much more time with hyper-descriptors than with conventional descriptors.

The use of hyper-descriptors allows to completely remove the classical filters used in standard IMAS algorithms to avoid problematic matches. Indeed, all post-processing filters that were usually applied in standard IMAS algorithms are now pointless:

1. Filtering repetitive matches (the *unique* filter step of Algorithm 6) is no longer useful. Algorithm 8 considers groups of close descriptors as one single hyper-descriptor holding all the information. A match between two hyper-descriptors is considered as a single match.
2. Filtering *one-to-multiple* matches is now naturally included by the fact that the IMAS-MATCHER of Algorithm 9 generalizes Lowe’s criterion [Low04] to hyper-descriptors.
3. *Multiple-to-one* matches are not forbidden with hyper-descriptors but do not appear any more in practice.

Algorithms 8 and 9 finally give birth to Algorithm 7, an IMAS algorithm based on hyper-descriptors. As simple as it is, this algorithm increases radically the quality of matches. No post-processing is needed in order to extract the underlying meaningful transformation. Thus, any parameter estimation approach like RANSAC [FB81], LO-RANSAC [CMK03], ORSA [MMM12,MMM16] or USAC [RCP+13], can be applied right after Algorithm 7. We then propose that any IMAS based on Algorithm 5 should evolve into Algorithm 7.

## 2.4 Two Structural and Computational Improvements

We describe in this section two computational tricks. The first one slightly modifies Lowe’s acceptance criterion in order to enable multiple matches for each hyper-descriptor. The second one is purely used for speed-up considerations, and consists in filtering flat descriptors in the early stages of the whole matching algorithm.

### 2.4.1 A *Contrario* Matching Revisited

In Lowe’s acceptance criterion, the ratio between the distance to the nearest neighbor and the distance to the second nearest neighbor is thresholded to decide if a match is accepted or rejected. The second nearest neighbor is taken in the target image. This has several drawbacks. First, it introduces a bias for small target images, which contain less descriptors and therefore pass the threshold more easily. A second structural bias is that this threshold also eliminates matches with repeated regions in the target images. One way of allowing one-to-multiple matches that are truly present in the target image is to create an *a contrario* model from an independent base of keypoints. We therefore propose to take a third image as background model, as it was first proposed in [CLM+08]. Instead of selecting the second nearest descriptors among those of the target image, Algorithm 10 uses the nearest hyper-keypoint among those of this third image.

The idea behind Algorithm 10 is consistent with Lowe’s justification. It evaluates on the *a contrario* image how likely it is that a descriptor matches so well just by chance.

By equipping Algorithm 7 with the IMAS-Matcher of Algorithm 10 we allow repetitions in an image to be recognized. In this way, more reliable information is passed on. For example, keypoints lying on repetitive windows in a building will not be removed, they will rather match with each other and add up when the meaningful transformation is queried while post-processing.

---

**Algorithm 10** IMAS-Acontrario-Matcher

---

**input:** three sets of  $\rho$ -hyper-descriptors  $D_1, D_2, D_a$ .

**parameters:** the match ratio  $\lambda \in ]0, 1[$ .

**start:**

$M = \emptyset$

▷ Storage of Matches

**foreach**  $d \in D_1$  **do**

$a \in \arg \min_{c \in D_2} \Delta(d, c)$

▷  $\Delta(x, y) = \min_{(\alpha, \beta) \in x \times y} \delta(\alpha, \beta)$

$b \in \arg \min_{c \in D_a} \Delta(d, c)$

**if**  $\frac{\Delta(d, a)}{\Delta(d, b)} \leq \lambda$  **then**

$M = M \cup \arg \min_{(\zeta, \eta) \in d \times a} \delta(\zeta, \eta)$

**return**  $M$

---

### 2.4.2 Filtering Flat Descriptors for Faster Computations

In order to speed-up the matching part of the algorithm, we can identify and eliminate unidirectional descriptors in the early stages of the algorithm. Flat descriptors are more likely to match each other and create too many false matches. These flat descriptors are identified with two internal filters in SIIM:

1. On-edge keypoints are considered as unstable. To detect an edge response, the ratio of smallest to largest principal curvatures of the DOG function (eigenvalues of the Hessian) is to be below a threshold. In our case, we set the threshold to 0.08 for octave scales less than 1 and to 0.06 otherwise.
2. Strongly biased descriptors towards a particular direction can also be eliminated by the means of the structure tensor. The eigenvalues of the structure tensor effectively summarize the predominant directions of the gradient around the keypoint. In our case, the ratio between the smallest and largest eigenvalues is set to be less than 0.06.

As argued in [Low04], there is no need to compute the eigenvalues themselves. Let  $\lambda_1$  and  $\lambda_2$ , be the eigenvalues of a the  $2 \times 2$  matrix  $Q$ . Then,  $\alpha = \frac{\lambda_1}{\lambda_2}$  must satisfy:

$$\frac{(\text{Tr } Q)^2}{\text{Det } Q} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \lambda_2} = \frac{(\alpha + 1)^2}{\alpha}. \quad (2.3)$$

The function  $f(\alpha) := \frac{(\alpha+1)^2}{\alpha}$  is an increasing function of  $\alpha$ . Therefore, thresholding the eigenvalue ratio of  $Q$  is equivalent to thresholding the ratio between the trace and the determinant of  $Q$ .

This filter is nonetheless optional and does not change significantly the final result. However, it often reduces the total computing time.

## 2.5 Numerical Results

Our IMAS method can be tested as an [IPOL demo](#) for two of the most popular state-of-the-art SIIMs, namely SIFT and SURF<sup>3</sup>. The IMAS versions of SIFT and SURF are now ensured to have minimal complexity thanks to the near optimal coverings described in Section 2.2.

---

<sup>3</sup>We use the SURF version developed in [OR15] which improves its former version in transition tilt tolerance.

Several versions of SIFT can also be tested in our IPOL demo. HalfSIFT [KSS07] and RootSIFT [AZ12] have been successfully applied in Computer Vision. They yield small modifications of SIFT descriptors but improve the quality of the results. By only taking the square root of a SIFT descriptor after a normalization, RootSIFT is known to outperform SIFT in terms of transition tilt tolerance, see Chapter 1. Unfortunately, most SIIMs fail in the case of non monotone intensity variations. HalfSIFT attempts to handle this by generating  $\bmod \pi$ -oriented descriptors. Indeed, this property makes HalfSIFT robust to contrast inversions. It improves the comparison of day/night images of the same objects, or images of the same objects taken in different wavelengths. Finally, a third descriptor, called HalfRootSIFT, cumulates the effects of RootSIFT and HalfSIFT. It is also available in the demo and improves over HalfSIFT.

### 2.5.1 Using Optimal Coverings

We first illustrate the gain obtained by using our near optimal coverings (see Chapter 1) instead of the classical coverings proposed in [MY09, PLYP12]. We refer the reader to Chapter 1 for a more rigorous approach. Table 2.2 shows a brief comparison between classical and optimal coverings for two SIIMs: SIFT + L1 norm and Root-SIFT + L2 norm.

**Table 2.2** Matching methods performance over query and target images from Figure 2.7. Computations were performed on an Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz with 4 cores.

M - Matches.

$ar$  - area ratio.

	M	$ar$	$ar^2$	Keypoints (seconds)	Matching (seconds)	Filters (seconds)
ASIFT + L1	801	13.7	189.6	6	36	0
Optimal ASIFT + L1	401	8.9	79.2	3	14	0
ARoot-SIFT + L2	821	13.7	189.6	6	10	1
Optimal ARoot-SIFT + L2	503	7.34	53.8	3	3	0



Figure 2.7: Graffiti.

Retrieved homographies in Table 2.2 were visually the same for all four methods. The mean homogeneous homography matrix  $H_{\text{mean}}$  and its standard deviation are shown in (2.4). As those statistics are difficult to interpret, we then compute in (2.5) the maximal distance of mapped points for all four homographies  $h_i$  with respect to the mean

homography  $h_{\text{mean}}$ . Equation (2.5) also indicates that the time saved in computations when using optimal coverings does not affect the accuracy of the retrieved homographies.

$$H_{\text{mean}} = \begin{bmatrix} 0.4356 & -0.6739 & 455.6987 \\ 0.4460 & 1.0201 & -51.2587 \\ 0.0005 & -0.0001 & 1.0000 \end{bmatrix}, \text{std} = \begin{bmatrix} 0.0011 & 0.0052 & 0.9886 \\ 0.0016 & 0.0061 & 0.3213 \\ 0.0000 & 0.0000 & 0 \end{bmatrix} \quad (2.4)$$

$$\max_{v \in \text{query image domain}} \max_{i \in 1, \dots, 4} \|h_{\text{mean}}(v) - h_i(v)\|_{L^2} = 3.2994. \quad (2.5)$$

## 2.5.2 Using Hyper-Descriptors

Using hyper-descriptors, introduced in Section 2.3, usually yields more quality matches than using standard descriptors. Descriptors that once were eliminated by the *multiple-to-one* / *one-to-multiple* filter are now kept without causing a burst of false matches. This means that no post-processing is needed after Algorithm 7. In order to identify the underlying meaningful transformations, we rely on four versions of the RANSAC Algorithm [FB81]: ORSA Homography [MMM12], ORSA Fundamental [MMM16] and USAC (Homography and Fundamental) [RCP<sup>+</sup>13].

We first highlight the need of all filters in the case of usual descriptors in Algorithm 5 and the advantage of Algorithm 7. Table 2.3 gives detailed information on how filters perform when applied sequentially from left to right. Optimal Affine RootSIFT was selected to perform all comparisons in this section.

It is usually required to remove repetitive matches when using RANSAC. Surprisingly, Table 2.3 (rows 3 and 4) shows an example in which applying the unique filter results at the end in a smaller quantity of true matches than not applying it. In practice, it is usually not the case and repetitive matches might produce a degenerate case for RANSAC, yielding at the end an inconsistent transformation. Table 2.3 (rows 1 and 2) shows that the application of multiple-to-one and one-to-multiple filters can be a more crucial step.

The last row of Table 2.3 shows that Algorithm 7 is already giving a clean set of matches and that most of the post-processing (Algorithm 6) is now pointless, except for the geometric filter. Figure 2.9 shows the output matches from Algorithm 5 and Algorithm 7 corresponding respectively to the second and last rows of Table 2.3. Figure 2.9 visually shows the improvement in terms of quality of Algorithm 7 over Algorithm 5.

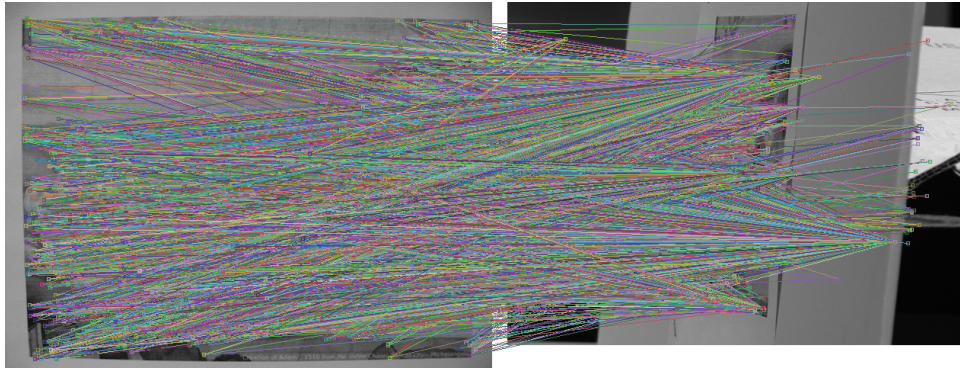
**Table 2.3** The first and second columns represent an IMAS algorithm and its output. From the third to the fifth columns one reads the sequence of filters appearing in Algorithm 6 applied (or not applied) to the output of the IMAS algorithm in question (in all cases configured for Optimal Affine-RootSIFT) which was run on the images from Figure 2.8. The final number of matches is coloured in **blue** if they are in accordance with the underlying homography; **red** on the contrary.

IMAS Algorithm	Output Matches	Unique Filter	Multiple-to-one and one-to-multiple Filters	ORSA Homography Filter [MMM12]
Algorithm 5	10986	-	-	<b>6</b>
Algorithm 5	10986	8864	-	0
Algorithm 5	10986	-	122	<b>42</b>
Algorithm 5	10986	8864	117	<b>38</b>
Algorithm 7	309	-	-	<b>98</b>

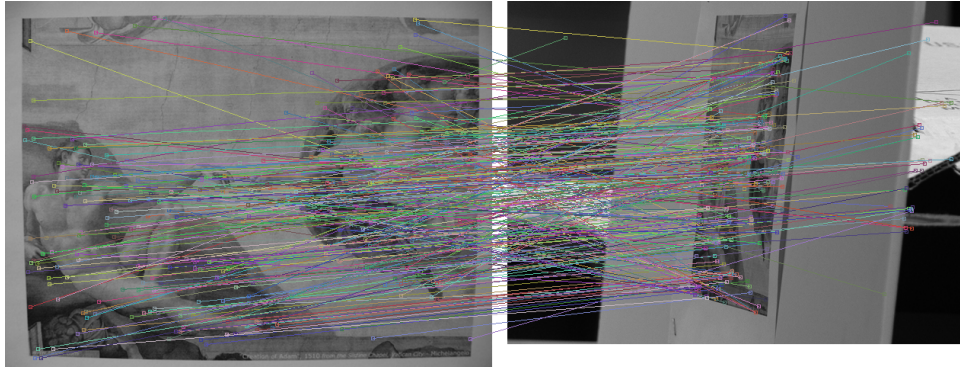




Figure 2.8: Adam.



(a) Algorithm 5 followed only by the unique filter of Algorithm 6, corresponding to the second row of Table 2.3. Too many *multiple-to-one* and *one-to-multiple* matches make it impossible for ORSA to determine the underlying homography.



(b) The raw output of Algorithm 7, corresponding to the last row of Table 2.3.

Figure 2.9: Analyzing Algorithms 5 and 7

### 2.5.3 Using the *A Contrario* Version of Lowe's Ratio Criterion

The *a contrario* matching in IMAS often incurs in a larger quantity of valid matches being accepted. However, the whole process relies on the hypothesis that the *a contrario* keypoints database represents an acceptable background model. This means that we should pay attention to the choice of the *a contrario* image. Usually, it is desired for this image to contain a wide variety of descriptors; natural images containing vegetation and water seem to go along with this property. Figure 2.10 has been selected for our experiments.





Figure 2.10: Selected a *contrario* image for our demo.

The interest of this *a contrario* version of Lowe’s acceptance criterion lies in two properties:

1. First, it has a tendency to increase the number of matches, without increasing the number of false matches. This is illustrated in the following by a panorama stitching experiment.
2. Second, it authorizes multiple matches per descriptor, hence the detection of structure even if it is repeated multiple times in the target images. This is illustrated in the following by the detection of repeated structures.

**Panorama stitching** Panorama stitching is the process of combining two or more images with overlapping regions from different viewpoints to produce a single panorama. If the homography relating two images is perfectly known, then each point in one image can be located with respect to the other image’s coordinates.

ORSA Homography [MMM12] can be applied to assess if a homography explains the output matches of Algorithm 7. If it exists, that homography contains all the information needed for retrieving the query image around the target one. Figure 2.11 shows 474 matches in accordance with the homography retrieved by ORSA [MMM12] applied right after Algorithm 7. Figure 2.12 shows a panorama stitching using this homography.

A slightly improved version of the above panorama stitching is obtained by introducing the *a contrario* matcher of Algorithm 10. The *a contrario* keypoints database was extracted from the *a contrario* image in Figure 2.10 by applying Algorithm 8. Resulting in 521 matches explained by the retrieved homography.

**Detection of repeated structures** As explained in Section 2.4.1, true repeated descriptors will annihilate each other if Algorithm 9 is applied. Figure 2.14 shows an example where most descriptors in the target image find a repeated copy somewhere in this very image. Only 19 matches were found in this scenario!

Figure 2.15 highlights the full potential of the *a contrario* matcher of Algorithm 10. This optional *a contrario* matcher is indeed well adapted when many repetitions are involved in the target image. Obviously its success depends on the choice of the *a contrario* image. On the other hand, this dependence is proved to be weak in practice (see Table 2.4).

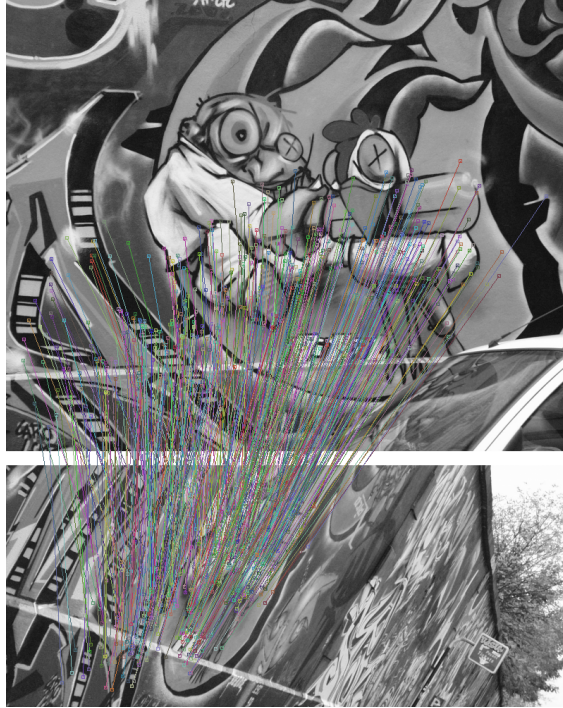


Figure 2.11: 474 matches among the output of Algorithm 7 were explained by ORSA Homography [MMM12].



Figure 2.12: Panorama stitching on Graffiti using the retrieved homography found by ORSA [MMM12].

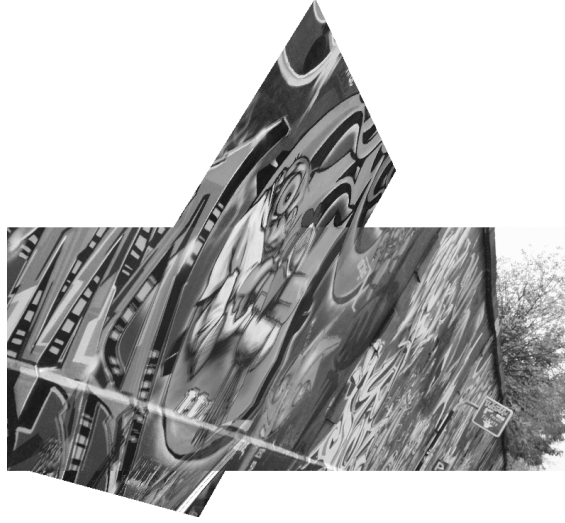


Figure 2.13: Panorama stitching on Graffiti with a *contrario* Matching. In this case, 521 matches among the output of Algorithm 7 with a *contrario* Matcher were explained by an homography retrieved by ORSA [MMM12].

**Table 2.4** Weak dependence on the *a contrario* image. The *a contrario* version of Algorithm 7 was applied on the images from Figure 2.15 with three different *a contrario* images shown in this table. The resulting output matches are varying around the 222 found matches in Figure 2.15, visually depending on the number of hyper-descriptors.


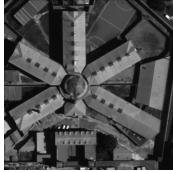

<i>A contrario</i> image	Size	Number of Hyper-Descriptors	Found Matches
	$640 \times 480$	1181	227
	$512 \times 512$	2601	212
	$668 \times 498$	2891	198



Figure 2.14: A total of 19 matches were found by Algorithm 7 with the matcher of Algorithm 9.





Figure 2.15: A total of 222 matches were found by Algorithm 7 (with the *a contrario* matcher of Algorithm 10). Comparing these results to those of Figure 2.14 shows the interest of the *a contrario* matcher.

# 3 Affine invariant image comparison under repetitive structures

## 3.1 Introduction

Everyday images are often composed of repeated objects, e.g. roof tiles, windows on buildings or chairs in a classroom. Humans not only identify these repetitions but also extract meaningful information from them. However, most of the state-of-the-art image matching algorithms still either fail to handle repetitions or were conceived not to treat them at all in order to be distinctive in practical applications [DMPC10].

The classic approach to image matching consists in three steps: detection, description and matching. First, key-points are detected in the compared images. Second, regions around these points are described and encoded in local invariant descriptors. Finally, all these descriptors are compared and possibly matched. Using local descriptors yields robustness to context changes. Both the detection and description steps are usually designed to ensure some invariance to various geometrical or radiometric changes. A large amount of research focused on using histogram representations, e.g. SIFT [Low04, ROD14], ASIFT [MY09], Shape Contexts [BMP02], Self-Similarity descriptors [SI07], etc. We refer the reader to [MS04, MTS<sup>+</sup>05, MP05] and Chapter 1 for in-depth comparative studies on image descriptors.

Although 3D viewpoint invariance seems quite utopian, its approximated version, affine invariance, has been widely studied in the literature [Lin93, Lin13b, MY09]. Chapter 1 also studies this subject, where the superiority of SIFT based descriptors for the latter invariance have been shown. On the other hand, Image Matching by Affine Simulation (IMAS) have been proven to be a reliable way to capture changes of point of view up to an impressive 88°, see [MY09, PLYP12, MMP15].

In order to be distinctive, most IMAS algorithms rely on the second-closest neighbor acceptance criterion proposed by D. Lowe in [Low04]. This criterion directly implies that the affine invariance property of these algorithms is strongly affected by repeated structures on the target image. To counteract these issues, Cao et al. [CLM<sup>+</sup>08] proposed two approaches to handle repetitions: the first is to compute the “second-closest neighbor” on an unrelated third image (where the repeated structure would not be present); the second is to add an *a-contrario* [DMM08] validation step, independent of the descriptor, which first selects a set of points around the key-points and then evaluates the agreement of gradient orientation on these points. Rabin et al. [RDG09] proposed an *a-contrario* validation for SIFT descriptor matches; the method requires learning the distribution of the descriptor space and uses the earth mover distance to quantify the descriptor similarity. Still a different *a-contrario* framework for match validation was described

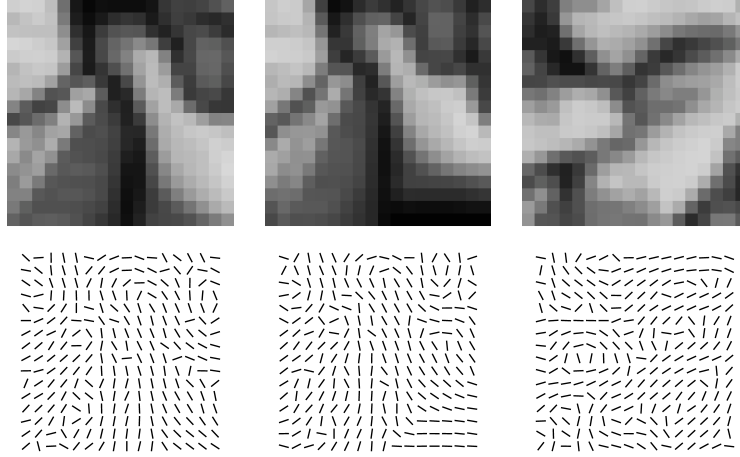


Figure 3.1: Three image patches and their corresponding orientation fields used as descriptors. The first two are similar while the third one is different.

in [GvGP15]; in this case it is based on comparing the gradient orientations in a patch and was suggested to use a local field of gradient orientations as a key-point descriptor.

However, none of these *a-contrario* methods is affine invariant. Here we follow the suggestion of [GvGP15], enriched by the IMAS approach, to build an *a-contrario* affine invariant key-point descriptor. Also we propose two variants of descriptor distances and the corresponding *a-contrario* models.

This chapter is organized as follows. Section 3.2 introduces the key-point descriptor based on a local field of image gradient orientation. The two *a-contrario* matchers for our descriptor are introduced in Sect. 3.3. Then, the IMAS techniques are explained in Sect. 3.4. Our experiments on images including repetitive structures, different viewpoint angles and noise are presented in Sect. 3.5.

## 3.2 The gradient angle field descriptor

The first step of the method is the key-point extraction. Each key-point comes with a position, scale and orientation. Then, a descriptor is associated to each key-point. A  $s \times s$  patch is extracted from the image centered at the position and orientation of the key-point. The sampling step is proportional to the key-point scale. Up to this point this is similar to the SIFT descriptor [Low04, ROD14]. But while the SIFT descriptor consists in a set of quantized histograms of the image gradient orientation, here we will follow the suggestion of [GvGP15] and use the actual values of patch gradient orientations as a descriptor. The descriptor of a key-point  $a$  will be then  $\alpha = \{\alpha_{ij}\}$ , where  $\alpha_{ij}$  are the angles of the gradient orientation at position  $i, j$  in the extracted patch of size  $s \times s = n$ . Figure 3.1 illustrates the idea. In all our experiments we used  $s = 20$  ( $n = 400$ ) and a sampling step of 1.5 relative to the key-point scale.

## 3.3 A contrario match validation

The proposed validation procedure is based on the *a contrario* theory [DMM08], which relies on the non-accidentalness principle [WT83, Low85]; informally, this principle states that there should be no detection in noise. In the words of Lowe, “we need to determine

the probability that each relation in the image could have arisen by accident,  $P(a)$ . Naturally, the smaller that this value is, the more likely the relation is to have a causal interpretation” [Low85, p. 39]. In our context, we need to assess the existence of a causal relation between two descriptors.

Given a pair of descriptors  $\alpha$  and  $\beta$ , a distance function  $d(\alpha, \beta)$  will be defined, together with a stochastic model  $\mathcal{H}_0$  for random descriptors used to evaluate accidentalness. We denote by  $D_{\mathcal{H}_0}$  a random variable (r.v.) corresponding to the distance between two random descriptors drawn from  $\mathcal{H}_0$ . To assess the accidentalness of a match  $(\alpha, \beta)$ , we need to evaluate the probability

$$\mathbb{P}[D_{\mathcal{H}_0} \leq d(\alpha, \beta)]$$

of observing under  $\mathcal{H}_0$  a distance  $D_{\mathcal{H}_0}$  smaller or equal than  $d(\alpha, \beta)$ . When this probability is small enough, there exists evidence to reject the null hypothesis and declare the match meaningful. However, one needs to consider that usually multiple pairs are tested. If 100 tests are performed, for example, it would not be surprising to observe an event that appears with probability 0.01 under random conditions. Thus, the number of tests  $N_T$  needs to be included as a correction term, as it is done in the statistical multiple hypothesis testing framework [GGQY07]. Following the *a contrario* methodology [DMM08], we define the *Number of False Alarms* (NFA) of a match as:

$$\text{NFA}(\alpha, \beta) = N_T \cdot \mathbb{P}[D_{\mathcal{H}_0} \leq d(\alpha, \beta)]. \quad (3.1)$$

Pairs with  $\text{NFA} \leq \varepsilon$ , for a predefined  $\varepsilon$  value, are accepted as valid matches. One can show [DMM08, P12] that under  $\mathcal{H}_0$ , the expected number of pairs with  $\text{NFA} \leq \varepsilon$  is bounded by  $\varepsilon$ . As a result,  $\varepsilon$  corresponds to the mean number of false detections per random image pair. In most practical applications the value  $\varepsilon = 1$  is suitable and we will set it once and for all.

An appropriate (unstructured) null hypothesis  $\mathcal{H}_0$  for random descriptors is that the gradient orientation angles are independent and isotropic. In other words, in a descriptor  $\Delta \in \mathcal{H}_0$ ,  $\{\Delta_{ij}\}$  is a family of independent random variables, uniformly distributed over  $[0, 2\pi)$ .

We will consider two distances, which will lead to two validation methods. The first one, denoted  $d^Q$ , is defined as the sum of quantized orientation errors:

$$d^Q(\alpha, \beta) = \sum_{ij} \mathbb{1}_{\left\{ \frac{|\text{Angle}(\alpha_{ij}, \beta_{ij})|}{\pi} > \rho \right\}}, \quad (3.2)$$

for a fixed orientation precision  $\rho \in (0, 1)$  (we use  $\rho = 0.3$ ). Given that the size of the descriptor is  $n$ , the value  $d^Q(\alpha, \beta) \in \{0, 1, \dots, n\}$ , with zero corresponding to a good match and  $n$  to the worst difference. This distance is similar to the one used in [CLM<sup>+</sup>08, sec.11.3]. The associated r.v.  $D_{\mathcal{H}_0}^Q$  corresponds to the sum of  $n$  independent Bernoulli random variables. Thus,

$$\mathbb{P}[D_{\mathcal{H}_0}^Q \leq d] = \sum_{k=0}^d \binom{n}{k} (1-\rho)^k \rho^{n-k} \quad (3.3)$$

is related to the tail of a Binomial distribution. We will denote AC-Q the method that uses the distance  $d^Q$ .

The second distance, denoted  $d^W$ , corresponds to a weighted sum of normalized orientation errors:

$$d^W(\alpha, \beta) = \sum_{ij} w_{ij} \frac{|\text{Angle}(\alpha_{ij}, \beta_{ij})|}{\pi}. \quad (3.4)$$



Now  $d^W(\alpha, \beta)$  is a real value between zero and  $\sum_{ij} w_{ij}$ . A perfect match has  $d^W(\alpha, \beta) = 0$  while the worst difference is  $d^W(\alpha, \beta) = \sum_{ij} w_{ij}$ . This is similar to the distance in [GvGP15] with the addition of the weights  $w_{ij}$ , which are used to impose a Gaussian window,

$$w_{ij} = \exp\left(-\frac{(i - s/2)^2 + (j - s/2)^2}{2\sigma^2}\right),$$

giving more relevance to the central points and requiring a more complex probability term than in [GvGP15]. The r.v.  $D_{\mathcal{H}_0}^W$  corresponds to the weighted sum of  $n$  independent and uniformly distributed random variables in  $[0, 1]$ . Using the vector index  $k$ , we have

$$\mathbb{P}[D_{\mathcal{H}_0}^W \leq d] = \mathbb{P}\left[\sum_k w_k e_k \leq d\right]$$

where the normalized errors  $e_k$  are  $U[0, 1]$ . The possible values of  $(e_1, \dots, e_n)$  can be seen as the points in a  $n$ -hypercube and the probability term is given by the volume of the intersection of the hypercube and the half-hyperspace  $\{(e_1, \dots, e_n) : \sum_k w_k e_k \leq d\}$ . There is a closed but complex formula for this volume [MM06]. For our purposes, however, it is enough to approximate it by the upper-bound given by the volume of the simplex  $\{(e_1, \dots, e_n) : e_k \geq 0, \sum_k w_k e_k \leq d\}$ ; thus

$$\mathbb{P}[D_{\mathcal{H}_0}^W \leq d] \leq \frac{1}{n!} \frac{d^n}{\prod_k w_k}. \quad (3.5)$$

We will denote AC-W the method that uses the distance  $d^W$ .

Finally, we need to specify the number of tests. Potentially, we may try to match any pixel of image  $I_1$  of size  $X_1 \times Y_1$  with any pixel of image  $I_2$  of size  $X_2 \times Y_2$ . We must also consider about  $\sqrt{X_1 Y_1}$  different patch orientations in  $I_1$  and  $\sqrt{X_2 Y_2}$  in  $I_2$ . To account for multiple scales, we consider  $\log_2(\max(X_1, Y_1))$  scales in  $I_1$  and  $\log_2(\max(X_2, Y_2))$  scales in  $I_2$ . As we will see, we perform several affine simulations leading to an extra factor  $\kappa$  per image (i.e. the area ratio from Chapter 1). All-in-all, the number of tests writes

$$N_T = (\kappa X_1 Y_1)^{\frac{3}{2}} \cdot \log_2(\max(X_1, Y_1)) \cdot (\kappa X_2 Y_2)^{\frac{3}{2}} \cdot \log_2(\max(X_2, Y_2)). \quad (3.6)$$

### 3.4 Affine invariance

As it will be shown in the following section, our methods are not initially affine invariant. Intuitively, the idea is to simulate a set of views from the initial images that will help to cover the affine space and then pairwise match those simulated images. The set of simulated views shall depend on concrete measurements of our methods' tolerance to viewpoint changes.

Most local descriptors and their corresponding matching methods are similarity-invariant. Unfortunately, slanted camera viewpoints (measured by  $t$  in Equation 1.1) will deteriorate the performance of almost any state-of-the-art matching method. To compensate this degradation at a minimum cost of complexity, we follow the ideas developed in Chapter 1 to compute optimal sets of affine simulations for each of our methods depending on the viewpoint tolerances, which we shall estimate in the next section. Under these conditions, Proposition 1.7 in Chapter 1 ensures that the constructed IMAS method is affine-invariant in practice. Indeed, there is at least one pair of simulated images whose viewpoint angle is not greater than the viewpoint tolerance of the matching method in question.

### 3.5 Experiments

The main objective of our two matchers is to allow repetitions to be captured. On the other hand, state-of-the-art descriptors are robust against noise, and Lowe’s second-closest neighbor criterion [Low04] is well known to render SIFT distinctive enough to be practical. All these properties are met for our methods even in the presence of viewpoint changes. A simple methodology is proposed to assess this claim.

The following procedure allows us to generate any number of test images  $u$  (query) and  $v$  (target) with the corresponding ground truth. Figure 3.2 shows an example. Let us consider three different and sufficiently distinct images,  $u_0$ ,  $v_0$ , and  $w_0$ . The test pair is generated randomly in four steps:

1. A  $N \times N$  patch is extracted from a random position in  $w_0$ ; a repetitive pattern  $P$  is composed by repeating the patch into a  $M \times M$  mosaic.
2. The pattern  $P$  is pasted into image  $u_0$  at a random position, producing image  $u_1$ ; similarly, the same pattern  $P$  is pasted in a random position of  $v_0$  to produce image  $v_1$ .
3. A random affine transform  $A$  is selected and used to optically simulate a distortion,  $v_2 = Av_1$ .
4. Finally, Gaussian noise is added to produce the final images,  $u = u_1 + n_u$  and  $v = v_2 + n_v$ .

Forcing  $A \in GL_*^+(2)$  in step 3 will incur in a change of point of view in  $v$  with respect to  $u$ ; the viewpoint angle can be selected.

This framework was used to compare systematically our methods to SIFT, RootSIFT and their affine invariant versions. Lowe’s criterion was applied for two match ratios (0.6 and 0.8). For each method and image pair, the *total number of true matches* and the corresponding *ratio of true matches* were computed. A match is considered as true if both constituting key-points lie inside the pattern  $P$  at the same position, modulo the repeated patch size. The displayed values are means after repeating the process for different generated image pairs. As the key-point extraction part is identical for all compared methods, the figures reflect only the performance of the descriptors and their matchers.

The frontal performance test of Table 3.1 confirms that our descriptors do handle repetitions and noise while still preserving a good ratio of true matches. Figures 3.3-3.4

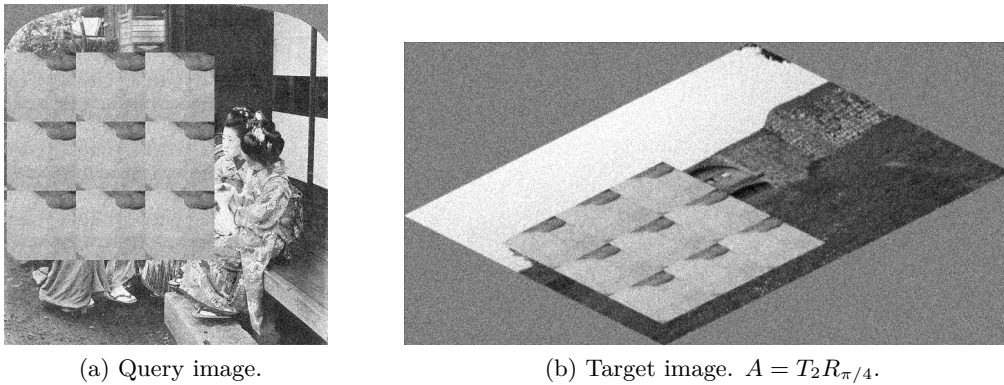


Figure 3.2: A generated image pair with repetitive structures.

**Table 3.1** Frontal performance test (i.e.  $A$  is a similarity). Mean values over 100 iterations.

Matching method	True matches	Ratio of true matches
SIFT L1 0.8	74	0.6577
SIFT L1 0.6	15	0.8054
RootSIFT 0.8	135	0.5383
RootSIFT 0.6	49	0.7812
AC-W	691	0.8679
AC-Q ( $\rho = 0.3$ )	1881	0.6261

**Table 3.2** Viewpoint tolerances (i.e. tilt tolerances from Chapter 1) obtained from the oblique performance test of Figure 3.4 with the convention that the ratio of true matches  $\geq 0.5$  and the total number of true matches  $\geq 10$ .

Matching method	Maximal viewpoint tolerance
SIFT L1 0.8	48°
SIFT L1 0.6	34°
RootSIFT 0.8	40°
RootSIFT 0.6	54°
AC-W	58°
AC-Q ( $\rho = 0.3$ )	54°

illustrates the benefits of our methods for varying viewpoint angle. Notice, however, the drastic fall in number and in ratio of true matches for all methods. Table 3.2 provides the estimated maximal viewpoint tolerances from the statistics presented in Figure 3.4; this brings to light a degradation in viewpoint tolerances (due to repetitive structures) for SIFT and RootSIFT with respect to results presented in Chapter 1 (respectively, 56° and 60°).

The theory of IMAS algorithms presented in Chapter 1 leads to optimal sets of affine simulations for each method, depending on viewpoint tolerances in Table 3.2. Figure 3.5 provides a geometrical representation of the optimal set of simulations for AC-W. Table 3.3 show the results for the affine invariant version of the methods in viewpoint angles from 60° to 80°. AC-W gets the overall best results; AC-Q produces significantly more good matches at the cost of a lower ratio of true matches.

Figures 3.6-3.7 show the benefits of the presented matchers: Affine AC-W and Affine AC-Q. Most matches from both methods are correct.

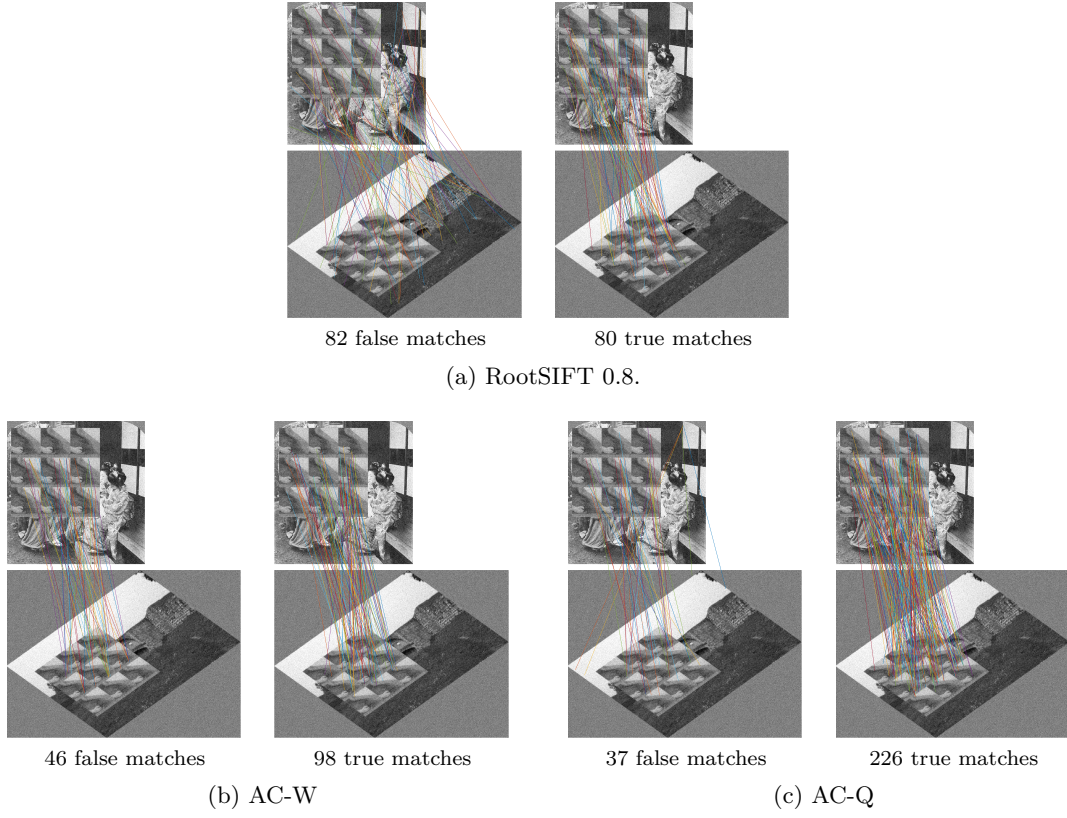


Figure 3.3: Performance of SIIM methods on a generated image pair for the oblique test.

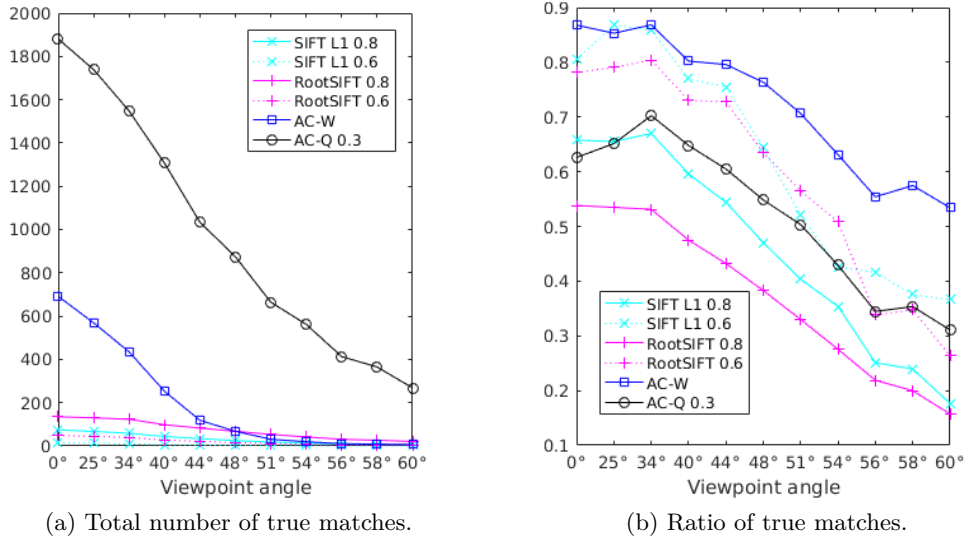


Figure 3.4: Oblique performance test. Each point represents the resulting mean over 100 iterations.

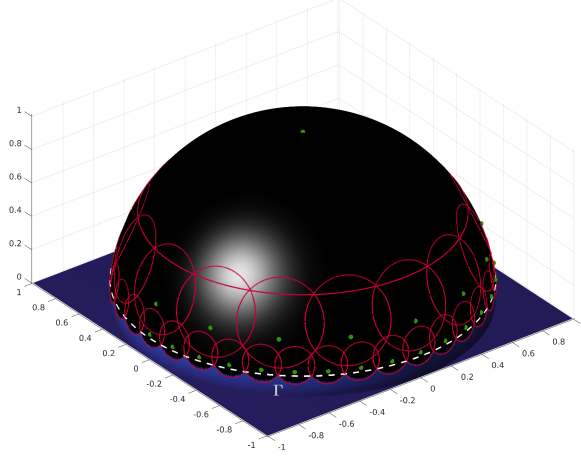


Figure 3.5: Optimal set of affine simulations for a method with viewpoint tolerances of  $58^\circ$ . Just 27 are enough to obtain an IMAS extension to  $80^\circ$ . Affine camera simulations (green); viewpoint tolerance from each simulation (red); visible viewpoints (black); maximal viewpoint tolerance for the IMAS method (dashed line).

**Table 3.3** Hard oblique performance test on affine invariant methods. The viewpoint angles are random and uniformly distributed between  $60^\circ$  and  $80^\circ$ . Mean values over 200 iterations.

Matching method	True matches	Ratio of true matches
Affine SIFT L1 0.8	33	0.4095
Affine SIFT L1 0.6	5	0.6463
Affine RootSIFT 0.8	48	0.2462
Affine RootSIFT 0.6	12	0.4934
Affine AC-W	195	0.7564
Affine AC-Q ( $\rho = 0.3$ )	913	0.2268



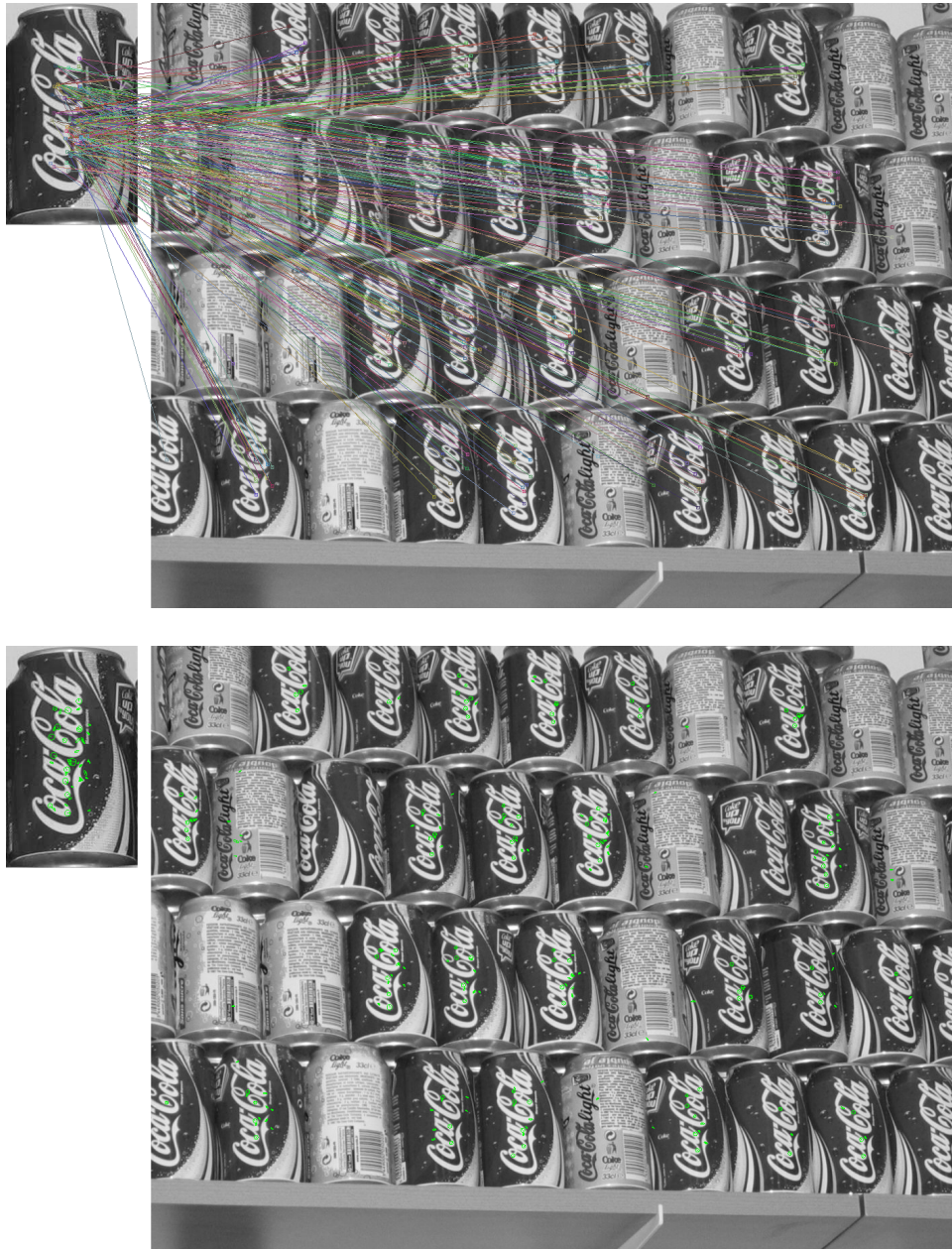


Figure 3.6: A total of 233 matches were found by Affine AC-W.



Figure 3.7: A total of 1059 matches were found by Affine AC-Q.

## 4 Conclusion

Image matching by affine simulations (IMAS) is acknowledged as one of the best methodologies to match images of the same scene regardless of the viewpoint change. Its time complexity is one of the main drawbacks that has been widely criticized in the literature. This drawback is mostly due to long time computations in the matching step. Indeed, the feature extraction complexity behaves linear on the number of keypoints; whereas the brute force matching complexity of these keypoints augments quadratically.

The mathematical derivations in Chapter 1 imply that IMAS based methods really are affine-invariant provided the base SIIM satisfies: scale+rotation invariance, sufficient distinctiveness, and an acceptable viewpoint tolerance measured by its *transition tilt*. We have proved that, as summarized in Figure 1.14, all former IMAS methods are over-simulating optical tilts. The procedure in the proof of Proposition 1.8 shows the way to generate optimal sets of simulations and we use it to present seven near optimal log $r$ -coverings ( $r \in [1.4, \dots, 2.0]$ ), some of them appearing in Figure 1.9. This led us to measure the tilt tolerance of several classic SIIMs to finally pair them with one of these seven optimal coverings.

We found for example that the optimal IMAS extension of SIFT needs twice less descriptors and therefore is four times faster than ASIFT [MY09, YM11]. This improvement applies to all state of the art IMAS, that can be accelerated by a factor of four. Another consequence is that the set of affine descriptors associated with an image can be halved.

Three concepts to improve affine invariant image matching have been presented in Chapter 2. First, in Section 2.2, we present in detail all the algorithms needed in order to optimize the set of simulations for generic IMAS algorithms depending on the SIIM and the targeted viewpoint tolerance. In Section 2.3 a more robust framework for IMAS algorithms was presented based on generalizations of standard keypoints and the second closest neighbor criteria introduced by Lowe [Low04]. Finally, the *a contrario* matching of Section 2.4.1 is an optional way of keeping true repetitive matches in images. However, the success of the *a contrario* matcher relies strongly on the assumption that our *a contrario* hyper-descriptors are able to capture the density of the space of natural hyper-descriptors.

For a fixed SIFT configuration, the final Optimal ASIFT from Chapter 2 differs from ASIFT [MY09, YM11] in:

1. *The set of affine simulations.* ASIFT generates highly redundant affine distortions whereas the final Optimal ASIFT applies a near optimal covering from Chapter 1.
2. *The structure.* ASIFT applies the SIFT method among all possible combinations of simulated query and target images; whereas the final Optimal ASIFT compares in a single stage both sets of hyper-descriptors coming from the query and target images.



**Table 4.1** Image matching performances on two viewpoint datasets. After matching each image pair, RANSAC-USAC [RCP<sup>+</sup>13] is run 100 times to measure its probability of success in retrieving corresponding ground truth homographies. Legend:  $N$  - the number of simulated affine maps on query and target;  $S$  - the number of successes (bounded by  $100 \times \boxed{\text{number}}$ ); the number of correctly matched image pairs; *inl.* - the average number of correct inliers; *AvE* - the average pixel error; *R* - the ratio of inliers/total. The  $\boxed{\text{numbers}}$  of image pairs in a dataset are boxed; *ET* - the average elapsed time in seconds. Hardware settings: (CPU) Intel(R) Xeon(R) W-2145 3.70GHz.

Matching method	N	EVD dataset [MMP15]						OxAff dataset [MTS <sup>+</sup> 05]					
		S	$\boxed{15}$	<i>inl.</i>	<i>AvE</i>	<i>R</i>	<i>ET</i>	S	$\boxed{40}$	<i>inl.</i>	<i>AvE</i>	<i>R</i>	<i>ET</i>
ASIFT [YM11]	41	750	9	129	4.9	0.42	7.07	4000	40	5697	1.9	0.98	13.53
Affine AC	25	300	3	70	4.0	0.95	7.19	3900	39	2723	2.7	0.92	27.97
Optimal Affine-SURF	25	221	3	74	3.8	0.37	6.92	3997	40	2283	2.6	0.84	7.88
Affine AC-W	25	300	3	45	4.0	0.98	13.46	3900	39	2698	2.7	0.92	42.44
Affine AC-Q	25	200	2	170	4.0	0.37	4.99	3900	39	2855	3.1	0.67	10.06
Optimal-Affine-RootSIFT	25	730	9	192	6.2	0.28	2.21	4000	40	2796	2.7	0.88	3.54
Optimal-Affine-RootSIFT revisited	25	498	5	291	6.1	0.34	2.82	4000	40	3191	2.8	0.79	4.62
AC	1	-	-	-	-	-	0.98	3400	34	775	1.6	0.99	1.77
AC-W	1	-	-	-	-	-	1.28	3400	34	774	1.6	0.99	2.40
AC-Q	1	-	-	-	-	-	1.20	3552	36	777	1.9	0.89	1.43
SURF	1	-	-	-	-	-	0.73	3898	39	809	2.1	0.79	0.61
RootSIFT	1	-	-	-	-	-	1.20	3900	39	1119	2.1	0.81	1.12

3. *Repetitive, but still significant, SIFT descriptors.* The final Optimal ASIFT method has the possibility of using the *a-contrario* matching revisited of Subsection 2.4.1 in order to match repetitive structures; ASIFT only uses the classic second nearest neighbour criteria which discards any repeated descriptor.
4. *Removal of flat descriptors.* The final Optimal ASIFT method uses the tensor structure to remove keypoints that will incur in one-dimensional descriptions (see Subsection 2.4.2) whereas ASIFT does not. This traduces in faster detection and matching steps.

In fact, all IMAS methods stemming from Chapter 2 will also benefit from these four improvements.

In Chapter 3 we described two SIIM methods for image comparison based on a new descriptor and two *a-contrario* matchers. These *a-contrario* matchers provide yet another way of replacing Lowe’s classic second nearest neighbour criterion. They are different from the *a-contrario* matching revisited of Subsection 2.4.1. The presented SIIM methods were tested in the presence of repetitive structures, noise and strong viewpoint differences, see Figure 3.3. Both methods have an excellent tolerance to viewpoint angle, comparable to the one provided by RootSIFT [Low04]. IMAS versions of them were created accordingly as proposed in Chapter 1-2. In our experiments the proposed IMAS methods produce better results than state-of-the-art methods in the presence of repetitive structures, strong viewpoints and noise. Future work will concentrate on combining our two SIIM methods in an attempt to get the best of both, a large number of true matches (provided by AC-Q) while keeping a high ratio against false ones (provided by AC-W).

As usual in image matching, each query descriptor is proposed to be matched with its

most similar target descriptor and later validated by the second nearest neighbor or other thresholding criteria. Instead, AC-W, AC-Q and Optimal Affine-RootSIFT revisited can match several (or none) target descriptors to each query descriptor without losing distinctiveness. However, these methods allowing to match repetitive structures are held back by the fact that classic geometric model estimators cannot cope with multiple matches between repetitive structures. Indeed, Figure 4.1 shows the result of applying RANSAC USAC [RCP<sup>+</sup>13] to those matches in Figure 3.7. Nevertheless, we combine these methods with RANSAC USAC [RCP<sup>+</sup>13] and show an experiment in Table 4.1 conducted on two well known datasets presenting strong viewpoint changes. All datasets include groundtruth homographies that were used to verify accuracy. First, local features were detected and matched, then RANSAC USAC [RCP<sup>+</sup>13] was applied and we declared a success if at least 80% of inliers (in consensus with the estimated homography) were in consensus with the groundtruth homography. In one hand, this experiment highlights the importance of affine simulation to improve the affine invariance of the presented descriptors. On the other hand, it shows a drop in performance of RANSAC USAC [RCP<sup>+</sup>13] if combined with matchers intended for capturing repetitive structures.

As it will be seen in Chapter 8, a band-aid solution to the above problem is provided by RANSAC<sub>affine</sub> from Chapter 6. Indeed, RANSAC<sub>affine</sub> looks for geometry inconsistencies to discard false matches between truly similar features. However, a more thorough analysis should take place to completely solve the aforementioned problem. Analyzing autosimilarities in images before matching them might be the key to the success. Indeed, the information on the emplacement of repeated objects in both query and target images might be valuable when matching them. This will be the focus of future work.



Figure 4.1: 29 matches, among those proposed by Affine AC-Q, were selected by RANSAC USAC [RCP<sup>+</sup>13] to be in consensus with the most predominant homography. Unfortunately, this predominant homography among Affine AC-Q matches is not providing any geometric information on the scene.

## Part II

# Learning the affine world



# 5 AID: an Affine Invariant Descriptor

## 5.1 Introduction

The classic approach to image matching consists in three steps: detection, description and matching [Low04]. First, keypoints are detected in both images to be compared. Second, regions around these points are described by local descriptors. Finally, all these descriptors are compared and possibly matched. Both the detection and description steps are usually designed to ensure some invariance to various geometric or radiometric changes. A benefit of local descriptors is that viewpoint deformations are well approximated by affine maps.

In Chapter 1, RootSIFT [AZ12] was reported to be the robustest descriptor to affine viewpoint changes (up to  $60^\circ$ ). To overcome this limitation, several simulation-based solutions have been proposed: ASIFT [YM11], FAIR-SURF [PLYP12], MODS [MMP15], Affine-AC-W in Chapter 3. Some optimal versions have been proposed in Chapter 2, including Optimal Affine-RootSIFT, which was proven to be the best choice.

On the other hand, local descriptors, which once were manually-designed, are currently being learned from data, with the promise of a better performance. Mimicking the classic process of image matching, they learn a similarity measure between image patches. In [ZK15], three similarity score architectures were introduced (CNN + a decision network). For stereo matching, two architectures based on CNNs were proposed in [ZL16], one of them computing the similarity score with the cosine proximity operator.

CNN-based geometric matching between images has also been tested for the case of affine and homography transformations [RAS18, DMR16]. In [RAS18], the POOL4 layer of the VGG-16 network [SZ14] was used for acquiring features from images and correlation maps fed to a regression network that outputs the best affine transform fitting the query into the target image. In a direct approach, the authors of [DMR16] trained a network to estimate the homography relating the query to the target image. Both [RAS18, DMR16] were trained on synthetically generated images, however neither of them took into account the blur caused by camera zoom-out or tilt.

In this chapter we combine manually-designed and learned methods in order to obtain a fast affine invariant image matching algorithm, capable of capturing strong viewpoint changes. The proposed method is based on the first stages of SIFT [Low04, ROD14], which ensure invariance to similarity transformations (translations, rotations and zooms) up to small perturbations (see [MY11] for a mathematical proof). At this point the SIFT descriptor is replaced by a neural network (Figure 5.3) that takes a  $60 \times 60$  patch as input and produces a 6272-element vector descriptor. The network is trained on a dataset containing pairs of patches related by affine transformations, aiming at producing similar descriptor vectors for affine pairs and dissimilar vectors otherwise [ZL16].

A simple way of measuring similarity between vector descriptors is through the cosine



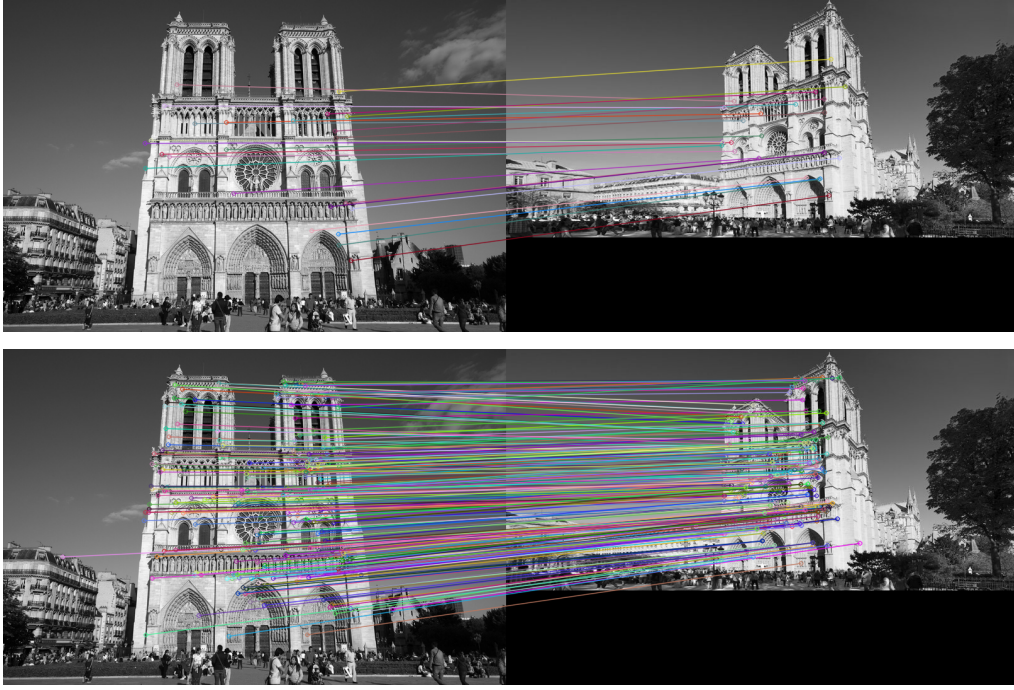


Figure 5.1: Top: matches by Optimal Affine-RootSIFT (48). Bottom: matches by the proposed SIFT-AID method (295).

proximity operator, i.e.  $\cos(\mathbf{x}, \mathbf{y}) := \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \|\mathbf{y}\|}$ . Therefore, we train the network to cluster similar descriptors with respect to angle. Finally, only the sign of each vector component is kept, leading to a binary descriptor. This allows to save memory and accelerate the matching process, while keeping the same level of performance and discriminative power. Figure 5.1 presents an example of the proposed method compared to the Optimal Affine-RootSIFT method.

## 5.2 Affine viewpoint simulation

A digital image  $\mathbf{u}$  obtained by any camera at infinity can be written as  $\mathbf{u} = \mathbf{S}_1 \mathbb{G}_1 A u_0$  where  $\mathbf{S}_1$  is the image sampling operator (on a unitary grid),  $A$  a linear map,  $u_0$  a continuous image and  $\mathbb{G}_\delta$  denotes the convolution by a Gaussian kernel broad enough to ensure no aliasing by  $\delta$ -sampling. Unfortunately,  $\mathbb{G}_1$  and  $A$  do not commute when  $A$  involves a tilt or a zoom. As a consequence, a simple warping  $A(\mathbf{u}_0)$  of the frontal image  $\mathbf{u}_0 := \mathbf{S}_1 \mathbb{G}_1 u_0$  is not a correct optical affine simulation of  $\mathbf{u}$ . As stated in Chapter 1, the correct way of simulating a tilt  $t$  in the  $x$ -direction is:

$$\mathbf{u} \rightarrow \mathbf{S}_1 T_t^x \mathbb{G}_{\sqrt{t^2-1}}^x I \mathbf{u},$$

where  $I$  is the Shannon-Whittaker interpolator and the superscript  $x$  indicates the operator takes place only in the  $x$ -direction. We denote  $\mathbb{T}_t^x := T_t^x \mathbb{G}_{\sqrt{t^2-1}}^x I$ . See Algorithm 4 for computing digital tilts in any direction.

It is clear that there is loss of information due to the blur; indeed, the operator  $\mathbb{T}_t^x$  is not invertible. Which means that, depending on the image  $\mathbf{u}$ , there might not be any optical transformation  $\mathbb{A}$  satisfying  $\mathbb{A}(\mathbf{u}_1) = \mathbf{u}_2$  or  $\mathbf{u}_1 = \mathbb{A}(\mathbf{u}_2)$ . Consider, for example,  $\mathbf{u}_1 = \mathbb{T}_t^x \mathbf{u}$  and  $\mathbf{u}_2 = \mathbb{T}_t^y \mathbf{u}$ .

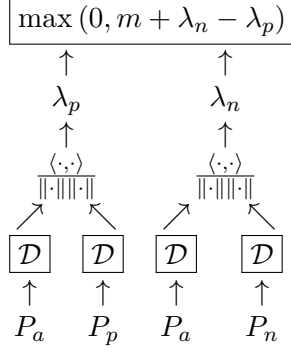


Figure 5.2: Diagram of the siamese network for training  $\mathcal{D}$ .

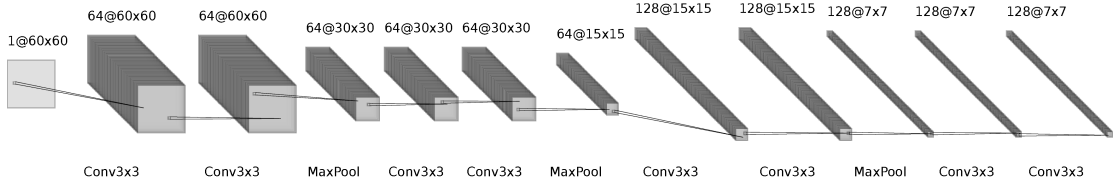


Figure 5.3: The proposed descriptor is computed using a CNN that produces a feature vector of dimension 6272.

With that in mind, we design a data generation scheme that, given an image  $\mathbf{u}$  and a pair of random affine transformations  $\mathbb{A}_1$  and  $\mathbb{A}_2$ , simulates affine views  $\mathbf{u}_1 = \mathbb{A}_1(\mathbf{u})$  and  $\mathbf{u}_2 = \mathbb{A}_2(\mathbf{u})$ . Both  $\mathbb{A}_1, \mathbb{A}_2$  with maximal viewpoint angles up to  $75^\circ$  with respect to  $\mathbf{u}$ . Instances of  $\mathbf{u}$  are provided accordingly from three independent MS-COCO [LMB<sup>+</sup>14] datasets for training, validation and test. Patch pairs seeing the same scene from  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are said to belong to the same *class* and will be used to train the descriptor network.

### 5.3 Descriptors and matching criteria

Inspired on [ZL16], our descriptor network  $\mathcal{D}$  is trained to produce similar descriptor vectors for patch pairs of the same class, and dissimilar vectors for patch pairs of different class. The network architecture is adapted from [DMR16], see Figure 5.3. It consists of 4 blocks of two convolutional layers each followed by batch normalization and ReLU activations. Between each block a max-pooling layer is introduced. A 2D Spatial Dropout with a probability 0.5 is applied after the last convolutional layer.

Here, dropout is not used to avoid over-fitting but to encourage the descriptor network to use all the dimensions of the feature vector. In addition, it does facilitate the learning process: the validation loss has proved to be much more stable than without dropout.

The affine approximation holds locally, which suggests the use of small patch sizes; on the other hand, small patches entail less information, leading to insufficient descriptions. As a compromise, we set the patch size to  $60 \times 60$ , which provides a good balance between locality and enough viewpoint information.



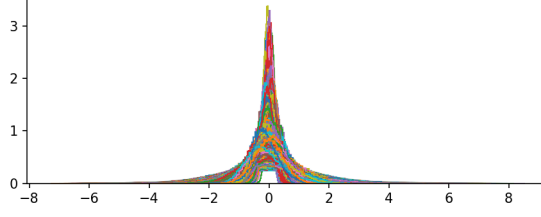


Figure 5.4: Density plots from each BigAID dimension (6272), computed over  $5 \cdot 10^4$  BigAID descriptions of random patches from the test dataset.

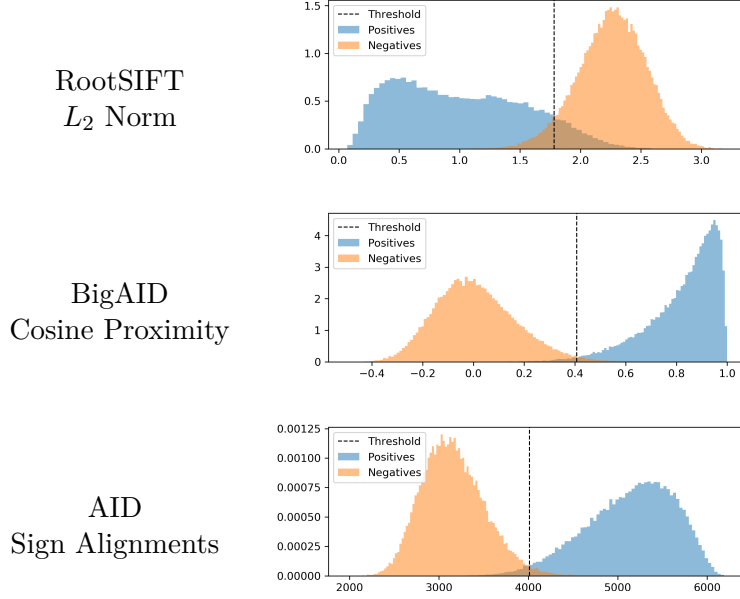


Figure 5.5: Positive and negative density estimation on measurements. For that,  $6 \cdot 10^5$  random intra and extra class pairs were used. The vertical line depicts the threshold minimizing both error probabilities: false negatives and false positives.

### 5.3.1 Training with hinge loss

During training, the descriptor network is immersed into a siamese network, represented in Figure 5.2. The siamese network consists of two identical sub-networks joined at the top by a virtual layer that computes the hinge loss between their two outputs:

$$\begin{aligned}\lambda_p &= \cos(\mathcal{D}(P_a), \mathcal{D}(P_p)), \\ \lambda_n &= \cos(\mathcal{D}(P_a), \mathcal{D}(P_n)),\end{aligned}$$

where patches  $P_a, P_p$  belong to the same class whereas  $P_n$  does not. While training, we extract  $P_n$  from a random image, different from the one used by  $P_a, P_p$ . We also simulate random contrast changes on all input patches. The hinge loss, i.e.

$$L(\lambda_p, \lambda_n) := \max(0, m + \lambda_n - \lambda_p),$$

is used with parameter  $m$  set to 0.2 in our experiments.

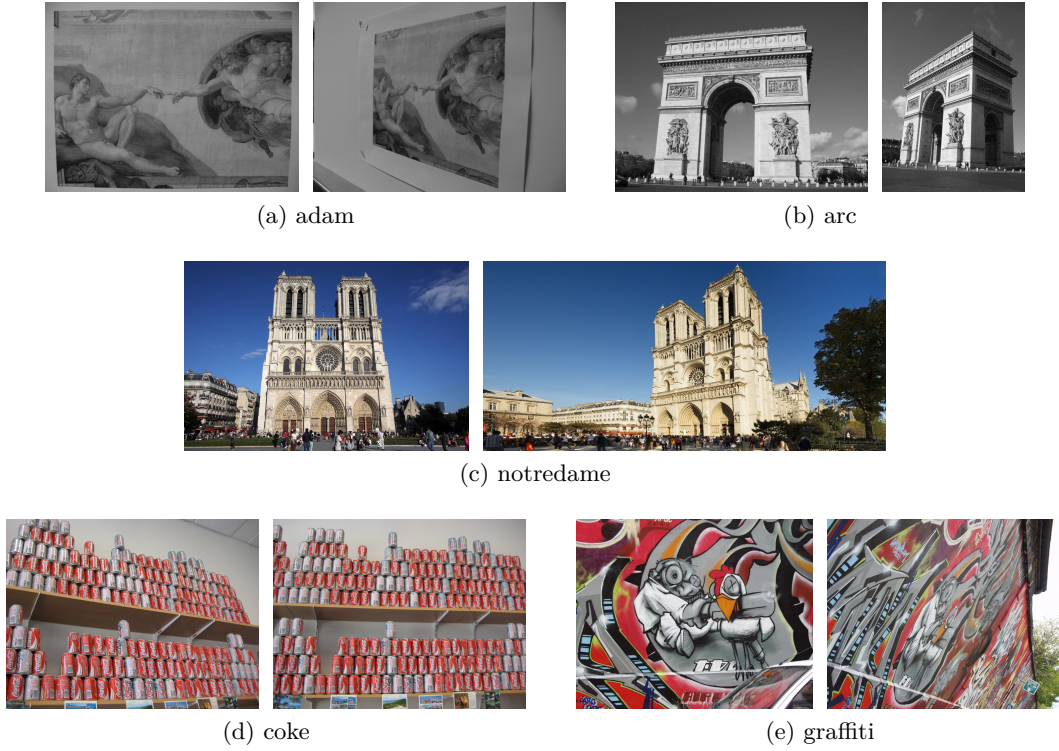


Figure 5.6: Viewpoint challenge dataset.

### 5.3.2 Binary descriptor and matching

When training is complete, the descriptor network is plugged out from the siamese network and expected to produce descriptors that capture affine invariant properties from input patches. We call this description *BigAID* (6272 floats). Figure 5.4 shows density estimations on each BigAID dimension. Notice the involvement of all the dimensions in the description and the symmetry of all densities around zero. With this in mind, we propose a new affine invariant descriptor, that we call *AID* (6272 bits), which only keeps the sign information from the BigAID. Two AID descriptors  $\mathbf{x}$  and  $\mathbf{y}$  are consequently matched via the sign alignment measure, i.e.

$$\sum_i \mathbb{1}_{\text{sign}(x_i)=\text{sign}(y_i)}.$$

Intra- and extra-class measure density estimations are shown in Figure 5.5 for Root-SIFT (128 floats = 4096 bits) and our descriptors, suggesting that for the BigAID and AID descriptors, a simple thresholding of their respective measures is sufficient to single out classes.

## 5.4 Experiments

Up until now, the descriptor network  $\mathcal{D}$  has only seen optically simulated input patches. Figure 5.6 provides a realistic viewpoint challenge dataset in the form of 5 pairs of images. Given a fixed set of SIFT keypoints from these images, the proposed methods are compared against RootSIFT in the section Test I of Table 5.1. The number of homography-consistent matches found by ORSA [MMM12] (an a-contrario validated RANSAC) shows

**Table 5.1** Viewpoint performance test. RS, A-RS, BigAID and AID denote Homography consistent Matches found by ORSA for RootSIFT, Optimal Affine-RootSIFT, BigAID and AID. The Second-Nearest-Neighbor ratio in RootSIFT and Optimal Affine-RootSIFT was set to 0.8. The thresholds for BigAID and AID were 0.4 and 4000, respectively. The star (\*) indicates on oracle keypoints.

	Test I: Using SIFT keypoints				
	# keypoints per image		Without viewpoint simulations		
	query	target	RS	BigAID	AID
coke	5443	5670	115	1316	<b>1409</b>
notredame	2285	1235	14	282	<b>295</b>
arc	1384	1387	40	<b>445</b>	420
graffiti	1661	3117	0	<b>182</b>	172
adam	269	192	30	67	<b>69</b>

	Test II: Using Optimal Affine-RootSIFT keypoints				
	# keypoints per image		With viewpoint simulations	Without viewpoint simulations	
	query	target	A-RS	BigAID*	AID*
coke	28609	31965	1395	5298	<b>5346</b>
notredame	11739	6444	48	590	<b>731</b>
arc	5719	4759	244	579	<b>600</b>
graffiti	14290	15225	<b>613</b>	502	516
adam	3647	2364	484	496	<b>520</b>

**Table 5.2** Time performance for Optimal Affine-RootSIFT, SIFT-BigAID and SIFT-AID. Elapsed time (in seconds) in building descriptors (ET-D) and matching them (ET-M); The star (\*) denotes GPU time.

	Optimal Affine-RootSIFT $L_2$ norm		SIFT-BigAID Cos. Prox.		SIFT-AID Sign Align.	
	ET-D	ET-M	ET-D*	ET-M	ET-D*	ET-M
coke	4.500	14.440	9.876	35.512	9.838	0.777
notredame	1.930	1.120	3.272	3.287	3.177	0.138
arc	1.520	0.380	2.581	2.236	2.465	0.107
graf	2.790	3.800	4.441	5.960	4.369	0.186
adam	1.210	0.130	0.601	0.088	0.525	0.030

the superiority of the AID descriptors with respect to RootSIFT. AID is more compact and has a similar performance to BigAID. For these reasons, we prefer the AID descriptor and we call *SIFT-AID* the matching method resulting from its combination with SIFT keypoints.

The A-RS column (Test II) in Table 5.1 shows the number of homography consistent matches for Optimal Affine-RootSIFT. Notice how SIFT-AID has comparable performances without using viewpoint simulations. But in some cases, it yields less matches, as for the *adam* pair. Why? As stated in Chapter 2, Optimal Affine-RootSIFT has about 7 times more keypoints than SIFT. Some of those keypoints come exclusively from simulated versions of the input images, i.e., they do not belong to the Gaussian pyramid of the original input images. To further test AID descriptors, we define an oracle yielding precise keypoints in the original Gaussian pyramid best approximating each keypoint from the first stages of Optimal Affine-RootSIFT. Keypoints provided by this oracle are the best possible choices that could have been found by the first stages of SIFT. Table 5.1 (Test II) also shows the number of homography consistent matches for oracle + AID descriptors. This experiment reveals that both AID and BigAID would have been sufficient to identify almost all Optimal Affine-RootSIFT matches, provided that proper keypoints had been correctly spotted by the first stages of SIFT. In the case of the *graffiti* pair, most of the missing matches for AID descriptors involve viewpoint angles close to  $75^\circ$ , the maximal viewpoint angle present in the training dataset.

Table 5.2 shows the time consumed by SIFT-AID and Optimal Affine-RootSIFT (as in Chapter 2) in building descriptors and matching them<sup>1</sup>. Overall, the SIFT-AID method can achieve results in less time than Optimal Affine-RootSIFT.

Finally, the simple classification process established in this chapter enables AID to be used for repetitive structures. Figure 5.7 shows raw matches from AID, validating its capacity to deal with repeated objects while staying distinctive.

---

<sup>1</sup>Hardware settings: (CPU) Intel(R) Core(TM) i7-6700HQ 2.60GHz; (GPU) NVIDIA Corporation GM204GLM [Quadro M5000M].



Figure 5.7: A total of 142 tentative matches were found by AID.



## 6 Robust estimation of local affine maps

### 6.1 Introduction

The problem of constructing affine invariant image descriptors by using an affine Gaussian scale space, which is equivalent to simulating affine distortions followed by the heat equation, has a long history starting with [Iij71, Blo92, Lin93, LG94]. The idea of affine shape adaptation was used as a basis for the work on affine invariant interest points and affine invariant matching in [LG94, Bau00, MS02, MS04, TVO99, TV04, TV00], including the Harris-Affine and Hessian-Affine region detectors [MS02, MS04]. Finally, the detectors MSER (Maximally Stable Extremal Region) [MCUP04] and LLD (Level Line Descriptor) [MSCG03, MSC<sup>+</sup>06, CLM<sup>+</sup>08] both rely on image level lines. Yet, the affine invariance of these descriptors in images acquired with real cameras is limited by the fact that optical blur and affine transforms do not commute, as shown in [MY09]. To overcome this limitation, the authors of [MY09] proposed to optically simulate affine transformations. This idea was also exploited in [PLYP12, MMP15], in Chapters 2-3 and more recently by the SIFT-AID method from Chapter 5, which combines SIFT keypoints with a CNN-based patch descriptor trained to capture affine invariance. Another recent possibility to obtain affine invariance is by learning affine-covariant region representations [MRM18], where a patch is normalized before description. The latter method together with the HardNet [MMRM17] descriptor was reported to be the state of the art in image matching under strong viewpoint changes for all detectors.

Image matching usually refers to estimating a global homographic transform between two images. An established approach [HZ03] consists in computing local image matches, which are then aggregated using the RANSAC (RANdom SAMple Consensus) algorithm [FB81] to estimate a homography. The same procedure is also used for fundamental matrix estimation.

Recently, CNN-based image matching approaches have been proposed for directly estimating global affine and homographic transformations [RAS18, DMR16]. In [RAS18], the POOL4 layer of the VGG-16 network [SZ14] was used for acquiring features from images and correlation maps fed to a regression network that outputs the best affine transform that fits the query to the target image. In a direct approach, the authors of [DMR16] trained a network to estimate the homography relating the query to the target image. Both [RAS18, DMR16] were trained on synthetically generated images, but neither of them took into account the blur caused by camera zoom-out or tilt.

The objective of this work is to improve image matching by refining two stages of its pipeline. The improvement of homography estimation can be accomplished, on the one hand, by increasing the number of keypoint correspondences as well as their accuracy, and on the other hand by improving the RANSAC aggregation step. The contributions

of this chapter, detailed below, address all these issues:

1. We propose a LOCAL Affine Transform Estimator (LOCATE) based on a neural network which estimates both the direct and inverse affine maps relating two patches, leading to a more accurate local geometry estimation.
2. To increase the number of correspondences we use the local affine information provided by LOCATE to guide the discovery of new candidates.
3. We introduce a reformulation of the consensus set (inliers) in RANSAC, incorporating the richer information provided by LOCATE, leading to an increase in the probability of success.

A prevalent element in this chapter is the LOCATE method, which yields a first-order approximation of the local geometry relating pairs of image patches, i.e, local affine maps or tangent planes, see Figure 6.1. The network architecture of LOCATE is a variation from the one in [DMR16] that provides a two-way estimation, which leads to an increase in robustness relative to the former network. Another difference with respect to [DMR16] is the use of affine simulated patches to train the networks. This simulation incorporates a realistic optical model that takes into account the blur caused by camera tilt and zoom [MY09]. This procedure allows to easily generate an arbitrarily large training set.

The affine information was already been used [FH15, FMH17] to predict location and pose from affine detectors like MSER [MCUP04], Harris-Affine [MS02] or Hessian-Affine [MS04]. We propose to complement the SIFT detector with a *guided matching* [HZ03] step that increases the number of correct matches by sampling new keypoints surrounding the initial ones. LOCATE’s accuracy in location, orientation and scale (i.e. rotation and position in the Gaussian pyramid) results in a drastic increase in the number of correspondences.

When estimating homographies from sets of correspondences with RANSAC, the use of first-order approximations allows to increase the performance in homography estimation. This has already been proposed in [RB16] by composing normalized affine maps provided by the Hessian Laplace detector. This detector can be replaced with Affnet [MRM18] since it has been shown to produce more accurate affine maps. The LOCATE method can be used as well for the same purpose. In addition, we propose a modification in the RANSAC consensus step. Instead of defining inliers only by location agreement, we also consider the agreement in tilt, rotations and scale of the local affine maps. We will show how these modifications improve homography estimation from a set of SIFT-like matches.

The rest of this chapter is organized as follows. Section 6.2 summarizes a formal methodology for simulating local viewpoint changes induced by real cameras, as required for training our network. The LOCATE method is introduced in Section 6.3. Section 6.4 and Section 6.5 present the proposed guided matching and our modified RANSAC step, respectively. The use of the proposed methods is illustrated with experiments in Section 6.6.

## 6.2 Affine Maps and Homographies

As stated in Chapter 1, a digital image  $\mathbf{u}$  obtained by any camera at infinity is modeled as  $\mathbf{u} = \mathbf{S}_1 \mathbb{G}_1 A u$ , where  $\mathbf{S}_1$  is the image sampling operator (on a unitary grid),  $A$  is an affine map,  $u$  is a continuous image and  $\mathbb{G}_\delta$  denotes the convolution by a Gaussian kernel broad enough to ensure no aliasing by  $\delta$ -sampling. This model takes into account the

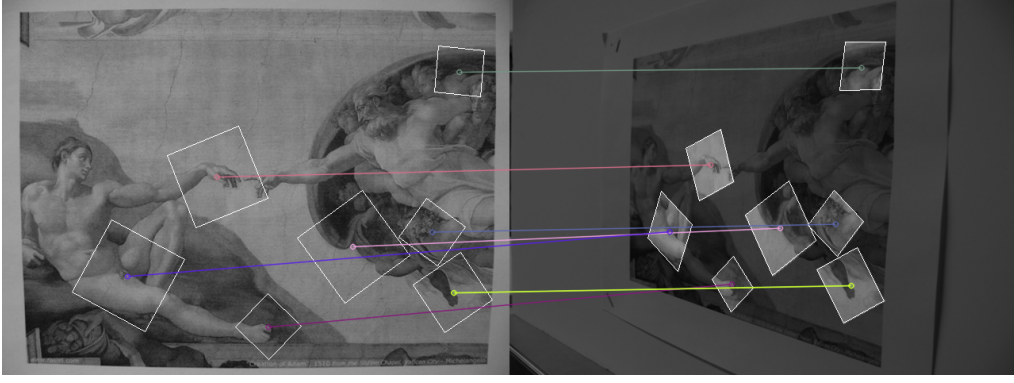


Figure 6.1: Some correspondences together with local affine maps estimated by the proposed LOCATE network. Patches on the target are warped versions of their corresponding query patch.

blur incurred when tilting or zooming a view. Notice that  $\mathbb{G}_1$  and  $A$  generally do not commute.

Proposition 1.1 from Chapter 1 claims that every  $A \in GL_*^+(2)$  is uniquely decomposed as in Equation 1.1,

$$A = \lambda R_1(\psi) T_t R_2(\phi),$$

where  $R_1, R_2$  are rotations and  $T_t = \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix}$  with  $t > 1$ ,  $\lambda > 0$ ,  $\phi \in [0, \pi)$  and  $\psi \in [0, 2\pi)$ . Furthermore, the above decomposition comes with a geometric interpretation (see Figure 1.1) where the longitude  $\phi$  and latitude  $\theta = \arccos \frac{1}{t}$  characterize the camera's viewpoint angles (or tilt),  $\psi$  parameterizes the camera roll and  $\lambda$  corresponds to the camera zoom. The so-called optical affine maps involving a tilt  $t$  in the  $z$ -direction and zoom  $\lambda$  are formally simulated by:

$$\mathbf{u} \mapsto \mathbf{S}_1 A \mathbb{G}_{\sqrt{t^2-1}}^z \mathbb{G}_{\sqrt{\lambda^2-1}} I \mathbf{u},$$

where  $I$  is the Shannon-Whittaker interpolator and the superscript  $z$  indicates that the operator takes place only in the  $z$ -direction. We denote by

$$\mathbb{A} := \mathbf{S}_1 A \mathbb{G}_{\sqrt{t^2-1}}^z \mathbb{G}_{\sqrt{\lambda^2-1}} I.$$

The operator  $\mathbb{A}$  is not always invertible and therefore its application might incur a loss of information. We refer to Chapter 5 for an example where no optical transformation  $\mathbb{A}$  is found between two views. With this in mind, we adopt the same data generation scheme proposed for training the affine invariant descriptors in Chapter 5. That is, given an image  $\mathbf{u}$  and a pair of optical affine maps  $\mathbb{A}_1$  and  $\mathbb{A}_2$ , we simulate affine views  $\mathbf{u}_1 = \mathbb{A}_1(\mathbf{u})$  and  $\mathbf{u}_2 = \mathbb{A}_2(\mathbf{u})$ . Our simulations involve maximal viewpoint angles of  $75^\circ$  with respect to  $\mathbf{u}$ . As in Chapter 5, the MS-COCO [LMB<sup>+</sup>14] dataset will provide instances of  $\mathbf{u}$  in training and validation. Patch pairs seeing the same scene from  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are said to belong to the same *class* and will be used to train the networks.

### 6.2.1 Local affine approximation of homographies

Let  $H = (h_{ij})_{i,j=1,\dots,3}$  be the  $3 \times 3$  matrix associated to the homography  $\eta(\cdot)$ . Let  $\mathbf{x}$  be the homogeneous coordinates vector associated to the image point  $x = (x_1, x_2)$  around



which we want to determine the local affine map. We denote by

$$y = (y_1, y_2) = \left( \frac{(H\mathbf{x})_1}{(H\mathbf{x})_3}, \frac{(H\mathbf{x})_2}{(H\mathbf{x})_3} \right) = \eta(x)$$

the image of  $x$  by the homography  $\eta$ .

The first order Taylor approximation of  $\eta$  at  $x$  leads to

$$\eta(x + z) = v + L(x + z) + o(\|z\|). \quad (6.1)$$

More specifically, if  $x = (0, 0)$ , we know that

$$y_i(z_1, z_2) = (h_{i1}z_1 + h_{i2}z_2 + h_{i3}) \left( \frac{1}{h_{33}} - \frac{h_{31}}{h_{33}^2}z_1 + -\frac{h_{32}}{h_{33}^2}z_2 + o(\|z\|) \right), \quad i = 1, 2.$$

Then, by polynomial identification in the Taylor formula

$$v + L(z) = \frac{1}{h_{33}} \begin{pmatrix} h_{13} \\ h_{23} \end{pmatrix} + \left[ \frac{1}{h_{33}} \begin{pmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{pmatrix} - \frac{1}{h_{33}^2} \begin{pmatrix} h_{13} \\ h_{23} \end{pmatrix} \begin{pmatrix} h_{31} & h_{32} \end{pmatrix} \right] \begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

where

$$\frac{1}{h_{33}} \begin{pmatrix} h_{13} \\ h_{23} \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}.$$

If  $x \neq (0, 0)$ , a simple change of variables  $z \rightarrow z + x$  would lead us back to the case  $x = (0, 0)$ . Notice that the resulting homography,

$$\tilde{\eta}(z) = \eta(z + x),$$

has an associated matrix determined by columns,

$$H_{\tilde{\eta}} = \left[ H \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad H \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad H \begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix} \right].$$

This brief computation shows that the vector  $v$  and the matrix  $L$  are determined through the following system of equations:

$$L = \begin{bmatrix} \frac{h_{11}-y_1h_{31}}{h_{31}x_1+h_{32}x_2+h_{33}} & \frac{h_{12}-y_1h_{32}}{h_{31}x_1+h_{32}x_2+h_{33}} \\ \frac{h_{21}-y_2h_{31}}{h_{31}x_1+h_{32}x_2+h_{33}} & \frac{h_{22}-y_2h_{32}}{h_{31}x_1+h_{32}x_2+h_{33}} \end{bmatrix}, \quad (6.2)$$

$$v = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - Lx. \quad (6.3)$$

This derivation allows us to compute the exact local affine approximation for a given homography. This will be useful for Section 6.5.1-6.5.2 and to assess the accuracy of our method when using annotated datasets.

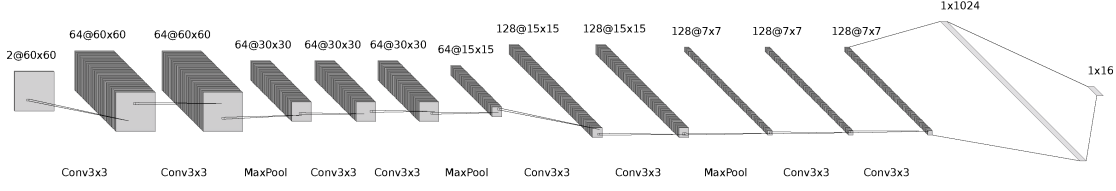


Figure 6.2: The proposed LOCATE network architecture. The last two layers are fully connected.

### 6.3 The Local Affine Transform Estimator

In this section we present the LOCAL Affine Transform Estimator (LOCATE) network whose architecture is adopted from [DMR16]. Unfortunately, the network as it is used in [DMR16] often incurs in wrong geometry estimates in the presence of strong blur or tilt, even when trained for this task. To address this issue, LOCATE estimates the affine transform that maps query to target *and* target to query. As it will be shown in Section 6.6, the simultaneous estimation of both, the direct and inverse maps, significantly improves the network performance.

The LOCATE architecture, shown in Figure 6.2, consists of 4 blocks of two convolutional layers each followed by batch normalization and ReLU activations. The first block receives as input two patches in the form of a two channel image. Between each block a max-pooling layer is introduced. A 2D spatial dropout with a probability 0.5 is applied after the last convolutional layer followed by 2 fully connected layers. The last layer outputs a vector of dimension 16, corresponding to the coordinates of eight points, the four transformed patch corners in both directions. We also tested a network trained to directly estimate the six parameters of local affine maps (translation plus the parameters in Equation 1.1) but we observed that this choice led to worse performances.

As argued in Chapter 5, the affine approximation holds locally, which suggests the use of small patch sizes; on the other hand, small patches contain less information, leading to insufficient geometry anchors. As a compromise, we set the patch size to  $60 \times 60$ , which provides a good balance between locality and sufficient viewpoint information.

#### 6.3.1 Training

The LOCATE network, as well as the network in [DMR16], were trained with data generated as described in Section 6.2; more specifically with pairs of patches belonging to the same class and involving small differences in translation, rotation and zoom, but possibly large tilts. The resulting networks will lead to an affine approximation of the exact transformation relating two observations. Both networks are trained from scratch until reaching a *plateau* for the loss in training and validation. While training we also simulate contrast changes on all input patches.

Let  $A_1, A_2$  denote two random affine maps and  $\mathbb{A}_1, \mathbb{A}_2$  their respective optical simulations. We assume  $A_1$  and  $A_2$  involve small perturbations in terms of similarity transformations. Let  $P_1$  and  $P_2$  be two square  $60 \times 60$ -patches simulated from a randomly chosen initial patch  $P$  by  $\mathbb{A}_1$  and  $\mathbb{A}_2$ , respectively. Let  $X = [x_1, x_2, x_3, x_4]$ , where  $x_i$  are the 2D coordinates of the four corners of a patch following a fixed order. We also define 4- and 8-point ground truth parameterizations respectively for the network [DMR16] and

the LOCATE network,

$$\begin{aligned} X^4 &:= A_1 A_2^{-1}(X), \\ X^8 &:= \left[ A_1 A_2^{-1}(X), A_2 A_1^{-1}(X) \right], \end{aligned} \quad (6.4)$$

where  $[\cdot, \cdot]$  denotes the concatenation of both vectors. Let  $\mathcal{N}^k$  be one of the presented networks with  $k$ -point parameterization. Then the loss is defined as sum of the Euclidean norm between corresponding points:

$$\sum_{i=1}^k \|\mathcal{N}^k(P_1, P_2)_i - X_i^k\|_{L_2}, \quad (6.5)$$

where the sub-index  $i$  denotes the  $i$ -th element of the vector.

### 6.3.2 From patches in the Gaussian pyramid to local affine maps

The training process described above allows the networks to be easily coupled with matching methods based on the SIFT [Low04] detector. Indeed, a SIFT-like patch is simply the square crop at the origin of some similarity transformation (translation, rotation and zoom) of the original image; additionally, patches corresponding to matched keypoints should suffer small similarity deformations but possibly strong tilts.

Consider two  $60 \times 60$ -patches,  $P_q$  and  $P_t$ , coming from the Gaussian pyramid of the query and target images, respectively. Let  $c_q$  and  $c_t$  be their centers expressed in image coordinates. Let also  $A_q$  be the affine map that converts from the query image domain to patch coordinates; likewise  $A_t$  converts from target to patch coordinates. Note that the affinities  $A_q$  and  $A_t$  are pure similarities, combining just the translation, rotation and zoom corresponding to the location, orientation and scale associated to SIFT-like keypoints. Finally, in order to locally approximate the transformation between query and target images (centered at  $c_q$  and  $c_t$ ), we only need the affine map relating  $P_q$  and  $P_t$ .

When fully trained, the presented networks are expected to predict the movements of patch corners. Let  $(x_i^q \leftrightarrow x_i^t)_{i=1,\dots,k}$  be a set of correspondences produced by one of the networks  $\mathcal{N}^k$ , where  $x_i^q$  and  $x_i^t$  denote query and target patch-coordinates, respectively, and  $k$ -point determines the point parameterization. Due to imprecisions in the prediction, these  $k$  correspondences are not necessarily related by an affinity. Then, the affine map  $A$  is estimated from the correspondences predicted by the network  $\mathcal{N}^k$  as the solution of the linear least squares problem

$$\min_A \sum_{i=1}^k \|Ax_i^q - x_i^t\|_{L_2}^2. \quad (6.6)$$

Finally, around  $c_q$ , the local affine map transforming the query into the target (in image coordinates) is

$$A_{q \rightarrow t} = A_t^{-1} A A_q. \quad (6.7)$$

We call LOCATE the method returning  $A_{q \rightarrow t}$  from the LOCATE network. Figure 6.3 visually shows estimated affine maps by the network [DMR16] (4 points) and LOCATE, as well as their respective incurred geometric errors. Four random patch pairs from the validation dataset (synthetic data) reveal the Achilles heel of network [DMR16]: zoom and translation. This visualization already justifies the use of the inverse information in the LOCATE method.

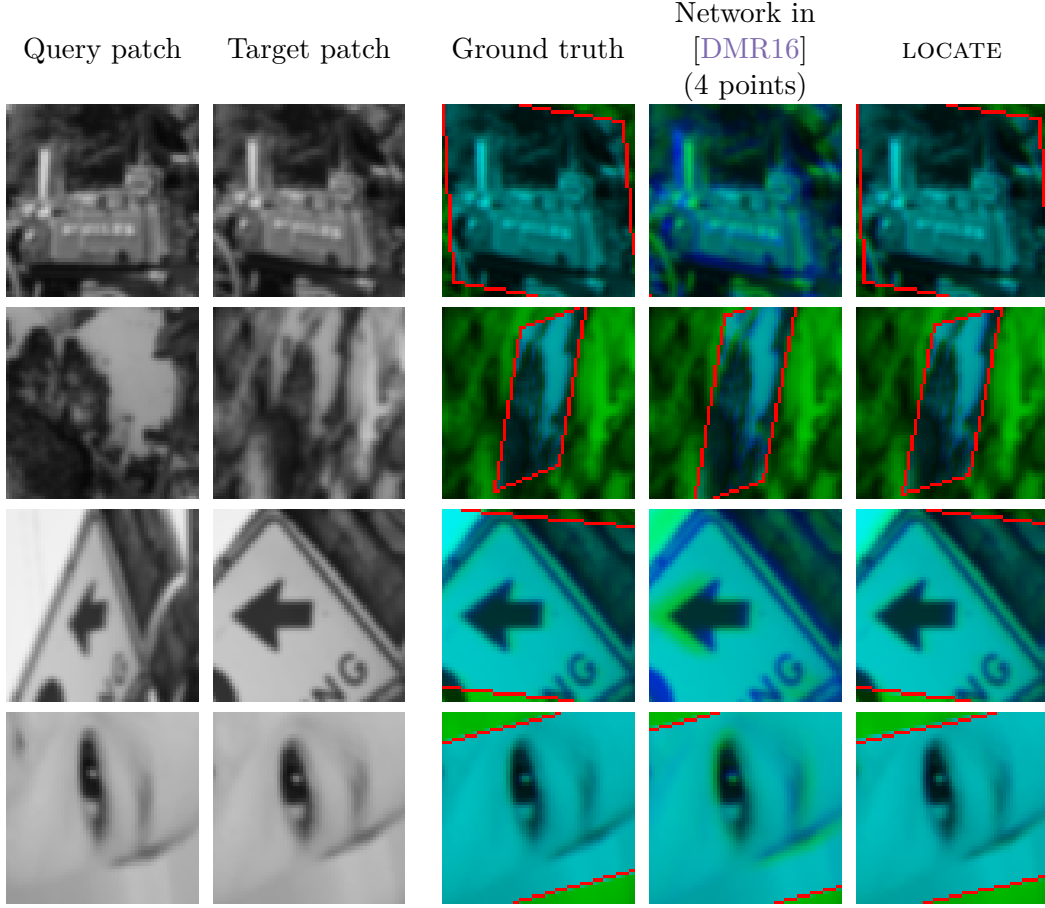


Figure 6.3: Four pairs of patches selected at random from the validation dataset and used as query and target input patches (columns 1-2). The three last columns show the drift error depicted by intense blue or intense green colors. Light blue means no error. Blue and green channels correspond to the target patch and a warped version of the corresponding query patch (the red line delimits its borders); the red channel is filled with zeros. 3rd column: groundtruth; 4th column: network in [DMR16] (4 points); 5th column: LOCATE network. Input patches are shown without contrast difference for clear visualization.

## 6.4 Refinement and Guided Matching

In this section, we present an iterative procedure that applies LOCATE to refine a set of existing matches, and then retrieves new ones by propagating the estimated local geometry. Think of the initial set of matches as correspondences resulting from a matching method, that includes both inliers and outliers. Each query and target keypoints have an associated location, orientation and scale (i.e. rotation and position in the Gaussian pyramid). The precise affine approximations between query and target obtained from LOCATE, allows to refine the matching by reducing the error in these three similarity parameters.

Furthermore, using the full affine transformations associated to the refined matches, allows to infer new match candidates by propagating the local geometry. The idea of propagating the local geometry from a set of matches was already proposed in the literature [FH15, FMH17]. In these cases the location and pose are derived from affine detectors like MSER [MCUP04], Harris-Affine [MS02] or Hessian-Affine [MS04]. Despite the fact that SIFT keypoints are more robust to similarities (see [RODM15]) than the previously

mentioned ones, no SIFT-like affine guided matching procedure was proposed yet. The reason for this is that the first method allowing to infer affine maps between SIFT-like patches is Affnet, which was very recently proposed. As we will see in Section 6.6, LOCATE reaches higher accuracy than Affnet. Therefore, in this chapter we introduce guided matching based on the LOCATE method.

The procedure is as follows. For each query keypoint from a refined match, four new keypoints are generated at the NE, NW, SE, SW corners of the query patch domain; see the four colored keypoints (red, green, light blue and blue) of Figure 6.4a. These points are then mapped into the target image domain (see Figure 6.4d) with rotations and positions in the target Gaussian pyramid inferred from the affine decomposition in Equation 1.1. These four pairs of points will represent new tentative matches, and each tentative match is validated by computing a similarity score between corresponding patches. For this task, we use the BigAID descriptor presented in Chapter 5 and the cosine proximity to measure the similarity.

This process can be iterated until some criteria is satisfied (e.g., a fixed number of iterations, the number of matches is stable, etc). In this chapter, we fix the number of iterations to 4. Each keypoint information is refined only once. To avoid redundancy, new matches falling nearby existing matches are removed (a threshold of 4 pixels was used). Therefore, any valid match proposal will cover new areas connecting the query and target images.

## 6.5 Robust Homography Estimation

The standard RANSAC algorithm computes the parameters fitting a mathematical model from observed data in the presence of outliers. Numerous improvements have been proposed in the literature for RANSAC, see [MMM12, MMM16, RCP<sup>+</sup>13, RFM<sup>+</sup>17], but the core idea remains the same.

In the case of homography estimation, the classic RANSAC algorithm returns the homography  $\eta_j$  computed in iteration  $j$  having the largest consensus of inliers among all iterations. The  $j$ -iteration can be described in two steps:

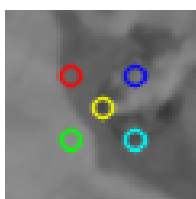
1. (Fitting) Randomly select  $s$  matches  $(x_i \leftrightarrow y_i)_{i=1,\dots,s}$  from the set of all matches ( $M_T$ ) and compute the homography  $\eta_j$  that yields the best fit.
2. (Consensus) Count the number of matches from  $M_T$  that are within a distance threshold of  $\kappa$  (i.e. counting inliers).

Notice that steps 1-2 only take into account point coordinates. From now on, we call this method *RANSAC*. With eight degrees of freedom for a homography matrix and each match defining two equations, this implies  $s = 4$ . The following subsections support the claim that incorporating the local affine information can further improve the performance of the RANSAC algorithm.

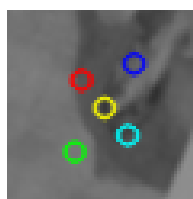
### 6.5.1 Homography fitting from local affine maps

From Section 6.2.1 we know how to locally approximate a homography by an affine map. Conversely, the problem of determining a homography from a set of affine maps at different locations was addressed in [BH16, RB16]. Let  $x \leftrightarrow y$  be a match and  $L = (l_{ij})_{i,j=1,2}$  the linear map in Equation 6.1. Then the unknown homography  $\eta$  must satisfy

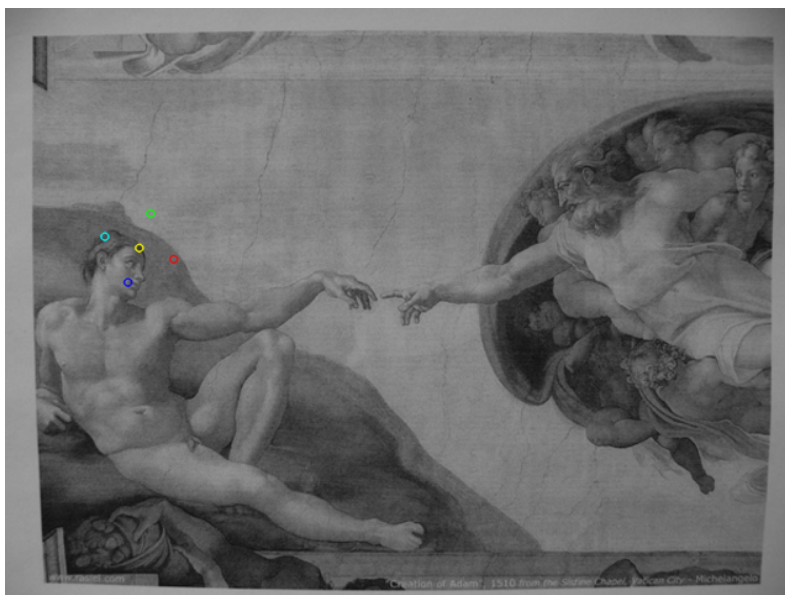
$$E_{6 \times 9} \cdot \vec{h} = \vec{0}, \quad (6.8)$$



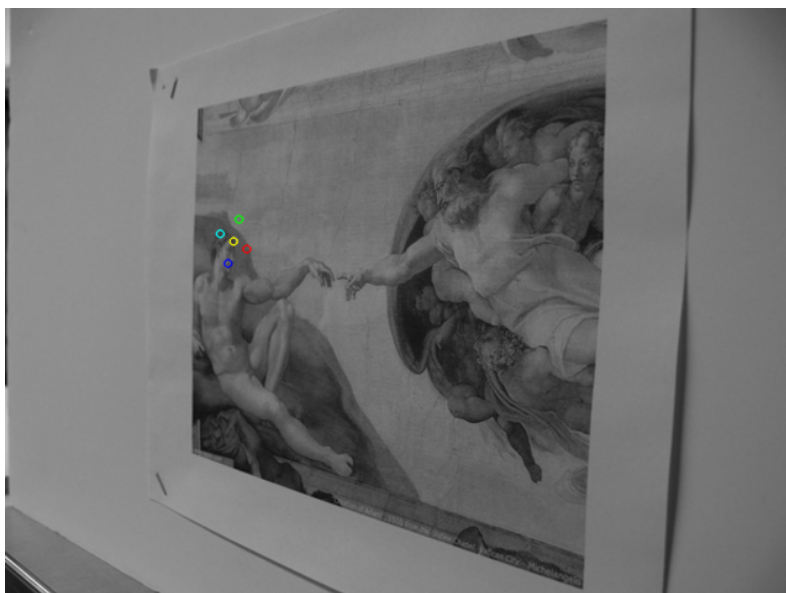
(a) Fixed query keypoints in the query patch.



(b) Inferred target keypoints by the LOCATE network in the target patch.



(c) Fixed query keypoints in the query image domain.



(d) Inferred target keypoints by the LOCATE method.

Figure 6.4: Four tentative matches around a match between the yellow keypoints.

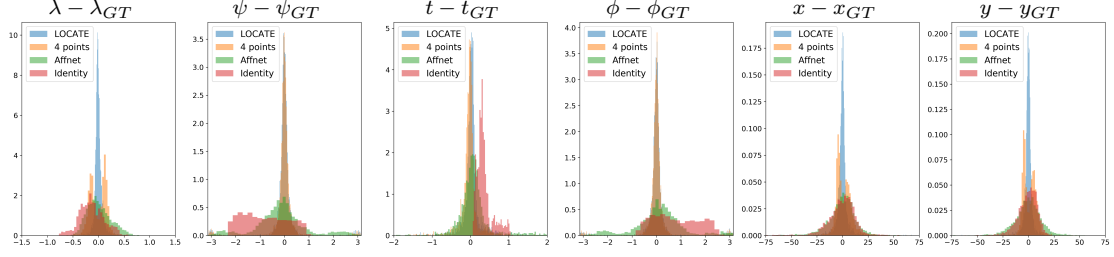


Figure 6.5: Affine error prediction in terms of the affine decomposition of Equation 1.1 (namely zoom  $\lambda$ , camera rotation  $\psi$ , tilt  $t$ , tilt direction  $\phi$ , and translation  $x, y$ ), for the proposed LOCATE method, the network [DMR16] (4 points), the Affnet method [MRM18] and the identity map method. The used dataset from Chapter 5 contains 3352 patch pairs with corresponding groundtruth. The sub-index  $GT$  means groundtruth, conversely, no sub-index stands for estimated parameters.

where  $E_{6 \times 9}$  is the matrix

$$\begin{bmatrix} 1 & & & -y_1 - l_{11}x_1 & -l_{11}x_2 & -l_{11} \\ & 1 & & -l_{12}x_1 & -y_1 - l_{12}x_2 & -l_{12} \\ & & 1 & -y_2 - l_{21}x_1 & -l_{21}x_2 & -l_{21} \\ & & & -l_{22}x_1 & -y_2 - l_{22}x_2 & -l_{22} \\ x_1 & x_2 & 1 & -y_1x_1 & -y_1x_2 & -y_1 \\ & & & x_1 & x_2 & 1 & -y_2x_1 & -y_2x_2 & -y_2 \end{bmatrix}, \quad (6.9)$$

and  $\vec{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^T$  is a vectorized version of the matrix  $H$  associated to  $\eta$ . The first four rows of  $E_{6 \times 9}$  are determined by Equation 6.2 and the last two are the classic equations derived from rewriting  $\eta(x) = y$  in terms of  $H\mathbf{x} = \mathbf{y}$ .

Clearly, two matches with their corresponding local affine maps can over-determine the homography matrix. Indeed, putting everything together provides with 12 equations

$$\begin{bmatrix} E_1 \\ E_2 \end{bmatrix}_{12 \times 9} \cdot \vec{h} = \vec{0},$$

where  $E_i$  denotes the matrix  $E$  appearing in Equation 6.8 for each match. To avoid the solution  $\vec{h} = \vec{0}$  we look for a unitary vector  $\vec{h}$  minimizing

$$\left\| \begin{bmatrix} E_1 \\ E_2 \end{bmatrix} \cdot \vec{h} \right\|,$$

see [HZ03] for more details.

We call  $RANSAC_{2pts}$  a RANSAC version in which the classic homography fitting of step 1 is replaced by the homography fitting of this section together with the LOCATE estimator. Note that  $RANSAC_{2pts}$  only needs two samples at each iteration ( $s = 2$ ).

### 6.5.2 Affine consensus for RANSAC homography

When matching two image patches, the transformation that relates them may not be consistent with the global transformation of the scene. This can be due to the presence of symmetric objects or even to failures in the matching process. For instance, suppose that two patches centered at the same scene location but with incoherent rotations are identified by a matching method. The symmetry issue is easy to address as usually we should have encountered as many keypoints as degrees of symmetry around the center;



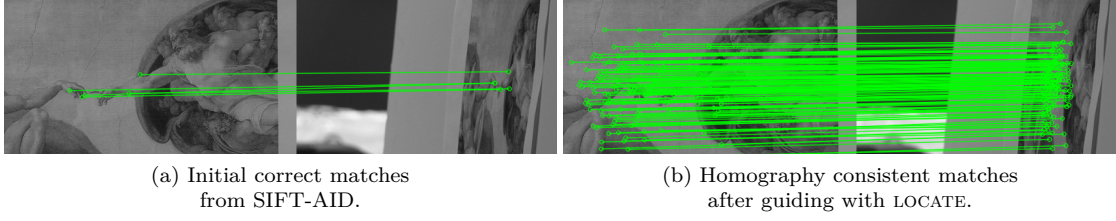


Figure 6.6: Guided matching for the adam pair, EVD [MMP15].

so at least two rotations will correspond. However, aberrant matches are not treated by the matching method nor by RANSAC. This problem can be circumvented by imposing consistency between the local affine maps and the proposed RANSAC model.

To impose local geometry consistency, most existing works [ZD06, MMP15] propose to measure the incurred error in mapping keypoints of a match  $x \leftrightarrow y$ , e.g.  $\|y - A(x)\| + \|x - A^{-1}(y)\|$ . Unlike them we propose to enforce geometry consistency directly on the transformations parameters given by Equation 1.1. In other words, we use the affine information to redefine the consensus set of a model.

Inliers are now defined as follows. Let  $A_E$  and  $A_H$  be, respectively, the estimated affine map by the LOCATE method and the testing affine map computed from the testing homography (using Equation 6.2). Let also  $[\lambda_E, \psi_E, t_E, \phi_E]$  and  $[\lambda_H, \psi_H, t_H, \phi_H]$  be, respectively, the affine parameters of  $A_E$  and  $A_H$ . We define the  $\alpha$ -vector between  $A_E$  and  $A_H$  as:

$$\alpha(A_E, A_H) = \left[ \max\left(\frac{\lambda_E}{\lambda_H}, \frac{\lambda_H}{\lambda_E}\right), \angle(\psi_E, \psi_H), \max\left(\frac{t_E}{t_H}, \frac{t_H}{t_E}\right), \angle(\phi_E, \phi_H) \right], \quad (6.10)$$

where  $\angle(\cdot, \cdot)$  denotes the angular difference. To test consistency between  $A_E$  and  $A_H$  we add to the classic threshold on the Euclidean distance, four more thresholds on the  $\alpha$ -vector. A perfect match would result in an  $\alpha$ -vector equal to  $[1, 0, 1, 0]$ . If we assume independence on each dimension, the resulting probability of a match passing all thresholds is the multiplication of individual probabilities. With this in mind, we claim that rough thresholds are enough to obtain good performances and that there is no need to optimize them. Thus, we propose to further refine inliers by accepting only those matches also satisfying

$$\alpha(A_E, A_H) < \left[ 2, \frac{\pi}{4}, 2, \frac{\pi}{8} \right], \quad (6.11)$$

where the above logical operation is true if and only if it holds true for each dimension.

We call  $RANSAC_{affine}$  the version of  $RANSAC_{2pts}$  that includes the affine consensus presented in this section.

## 6.6 Experiments

To the best of our knowledge, the most suitable and effective means of estimating affine maps connecting two patches are: Affnet [MRM18], the network [DMR16], and now the LOCATE method. The procedure described in Subsection 6.3.2 works for both networks: [DMR16] and LOCATE. On the other hand, Affnet was conceived to predict normalizing ellipse shapes for single patches based on a 3-variable parametrization. The connection provided by two Affnet-normalizing affine maps for the query and target patches is richer than each normalizing transformation. Indeed, for different choices of  $A_1 = T_1 R_1$  and



**Table 6.1** Guided matching and refinement performances on three viewpoint datasets with seed correspondences from two affine invariant matching methods: SIFT-AID from Chapter 5 and SIFT-Affnet [MRM18]-HardNet [MMRM17] (SIFT-Affnet). After refinement and guiding on each image pair, RANSAC-USAC [RCP<sup>+</sup>13] is run 100 times to measure its probability of success in retrieving corresponding ground truth homographies. Legend: S - the number of successes (bounded by  $100 \times \boxed{\text{number}}$ ); the number of correctly matched image pairs; inl. - the average number of correct inliers; AvE - the average pixel error; R - the ratio of inliers/total. The  $\boxed{\text{numbers}}$  of image pairs in a dataset are boxed.

Matching method	Guiding affine map	SIFT-AID dataset from Chapter 5					EVD dataset [MMP15]					OxAff Viewpoint dataset [MTS <sup>+</sup> 05]				
		S	$\boxed{5}$	inl.	AvE	R	S	$\boxed{15}$	inl.	AvE	R	S	$\boxed{10}$	inl.	AvE	R
SIFT-AID	None	500	5	508	6.2	0.24	100	1	162	6.2	0.11	1000	10	1840	4.1	0.43
	Identity	487	5	114	6.3	0.33	100	1	19	6.9	0.46	1000	10	1546	4.6	0.62
	LOCATE	500	5	<b>1438</b>	5.2	<b>0.44</b>	200	2	<b>862</b>	<b>3.8</b>	0.49	1000	10	<b>7198</b>	<b>2.7</b>	<b>0.71</b>
	4 points	500	5	1166	<b>5.1</b>	0.41	200	2	548	4.1	0.46	1000	10	6725	2.8	0.70
	Affnet	487	5	328	7.0	0.31	103	2	142	6.7	<b>0.50</b>	1000	10	2223	5.4	0.57
SIFT-Affnet	None	400	4	99	<b>3.8</b>	<b>0.79</b>	235	3	13	7.9	0.64	1000	10	1185	<b>2.1</b>	<b>0.96</b>
	Identity	300	3	32	4.2	0.71	0	0	0	-	-	895	9	1336	3.5	0.94
	LOCATE	400	4	<b>620</b>	4.7	0.72	200	2	151	5.6	<b>0.98</b>	1000	10	<b>6871</b>	2.5	<b>0.96</b>
	4 points	400	4	448	4.6	0.73	101	2	<b>169</b>	<b>3.1</b>	0.94	1000	10	6164	2.7	0.94
	Affnet	400	4	78	5.8	0.69	100	1	28	5.6	0.86	1000	10	1724	4.8	0.88

$A_2 = T_2 R_2$  one would need the four parameters (zoom, camera rotation, tilt and tilt direction) in Equation 1.1 in order to express  $A_2 A_1^{-1}$ . However, Affnet does not estimate translations. We claim that the LOCATE method out-performs the other two state-of-the-art methods in terms of precision.

Please note that the networks were trained exclusively with simulated patches, let us now try on real patches. The passage from affine cameras to real cameras is a big gap to fill by both [DMR16] and LOCATE networks. We expect them to generalize the affine world to all sorts of geometry as long as the Taylor approximation holds.

### 6.6.1 Does precision really matter?

As a first evaluation of the precision in a realistic environment we used the viewpoint dataset from Chapter 5, consisting of five pairs of images with their groundtruth homographies and 3352 true matches. Notice that Equations 6.2-6.3 allow us to compute groundtruth local affine maps around each match. Figure 6.5 shows the accuracy of Affnet [MRM18], the 4 points network [DMR16] and LOCATE, represented by error density functions with respect to the affine decomposition appearing in Equation 1.1. Ideally, we expect a Dirac delta function centered at 0 for a perfect method. This is approximately true for the LOCATE method. The experiment also illustrates the failure of the network [DMR16] in predicting zoom and translation (as shown in Figure 6.3). Note in Figure 6.5 that translations from the Affnet [MRM18] method do not quite match those from the Identity method; this difference can be explained by the connecting mapping itself, see Figure 6.7, as

$$A_{1 \rightarrow 2}(\mathbf{x}) = A_2 \left( A_1^{-1} \mathbf{x} - A_1^{-1} \mathbf{c} \right) + \mathbf{c}$$

**Table 6.2** Homography estimation performances for RANSAC, RANSAC<sub>2pts</sub> and RANSAC<sub>affine</sub> for three matching methods: RootSIFT [AZ12], SIFT-AID from Chapter 5, and SIFT-Affnet [MRM18]-HardNet [MMRM17] (SIFT-Affnet). Each RANSAC ran for 1000 internal iterations. To measure probability of success, all RANSACs were run 100 times on resulting matches from each pair of images. Legend: S - the number of successes (bounded by  $100 \times \boxed{\text{number}}$ ); the number of correctly matched image pairs; inl. - the average number of correct inliers; AvE - the average pixel error. The  $\boxed{\text{numbers}}$  of image pairs in a dataset are boxed.

Matching method	Homography Estimator	EF dataset [ZR11]				EVD dataset [MMP15]				OxAff dataset [MTS <sup>+</sup> 05]				SymB dataset [HS12]			
		S	$\boxed{33}$	inl.	AvE	S	$\boxed{15}$	inl.	AvE	S	$\boxed{40}$	inl.	AvE	S	$\boxed{46}$	inl.	AvE
RootSIFT	RANSAC	2403	26	51	3.2	0	0	0	-	3806	39	580	1.2	2693	31	102	2.8
	RANSAC <sub>2pts</sub>	2633	28	46	3.7	0	0	0	-	3893	39	566	1.2	3219	34	84	3.3
	RANSAC <sub>affine</sub>	<b>2805</b>	<b>30</b>	28	3.4	0	0	0	-	<b>3899</b>	<b>39</b>	404	1.1	<b>3297</b>	<b>36</b>	54	3.4
SIFT-AID	RANSAC	879	23	78	6.6	82	1	40	7.8	3600	39	1477	4.8	1014	19	450	6.8
	RANSAC <sub>2pts</sub>	1829	27	84	6.1	99	1	72	6.3	3917	40	1459	4.5	1867	30	327	6.5
	RANSAC <sub>affine</sub>	<b>1996</b>	<b>30</b>	39	5.8	<b>166</b>	<b>5</b>	37	8.2	<b>3939</b>	<b>40</b>	852	4.0	<b>2341</b>	<b>38</b>	138	6.6
SIFT-Affnet	RANSAC	2475	25	47	3.7	200	2	16	8.0	<b>4000</b>	<b>40</b>	805	2.3	2999	31	108	3.5
	RANSAC <sub>2pts</sub>	2707	28	43	3.6	<b>300</b>	<b>3</b>	10	7.6	<b>4000</b>	<b>40</b>	805	2.3	3268	34	99	3.4
	RANSAC <sub>affine</sub>	<b>2826</b>	<b>29</b>	29	3.5	200	2	12	7.4	<b>4000</b>	<b>40</b>	562	2.2	<b>3285</b>	<b>36</b>	65	3.5

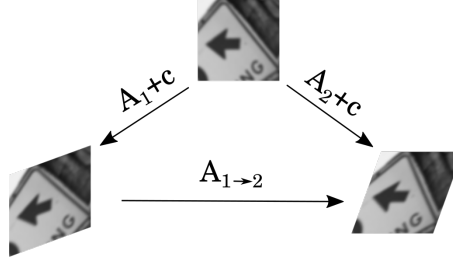


Figure 6.7: Passage from Affnet affine maps ( $A_1, A_2$ ) to the connecting mapping  $A_{1 \rightarrow 2}$ . The center of the normalize patch (on top) corresponds to the origin in normalized coordinates.

is different from  $A_2 A_1^{-1} \mathbf{x}$ , where  $\mathbf{c}$  denotes the center of patch domain and  $A_i$  are the estimated affine maps by Affnet. LOCATE, with the only addition of tracking points movements associated to the inverse affine map, obtains better result than [DMR16]. As expected, both [DMR16] and LOCATE perform better than Affnet [MRM18]. Indeed, Affnet analyzes one patch at a time, whereas [DMR16] and LOCATE have access to both patches simultaneously. However, in practice, using Affnet involves less computations.

The following experiment shows that the precision improvement of LOCATE indeed results in better guided image matching performance. Table 6.1 shows that LOCATE has the overall best performance of all methods. LOCATE usually boost the number of inliers as well as the ratio of inliers while always being the lowest or close to lowest average pixel error. By construction, this boost in inliers means that new areas are connected between the image pairs, see Figure 6.6 for an example. Moreover, the probability of success of RANSAC USAC [RCP<sup>+</sup>13] is not diminished with respect to the matching method itself, this is observed in the “None” rows of Table 6.1. We remark the capacity of our guided matching method to expand true matches while keeping the number of false matches low.

### 6.6.2 Can RANSAC<sub>affine</sub> improve homography estimation?

In the previous paragraphs we established the precision of the local affine maps provided by the LOCATE method. We now focus on the evaluation of the three variants of RANSAC. In order to highlight the benefits of local geometry in estimating homographies,

we drop all the improvements in RANSAC USAC [RCP<sup>+</sup>13] and head back to the base RANSAC. But notice that most improvements proposed in RANSAC USAC [RCP<sup>+</sup>13] can also be applied to RANSAC<sub>2pts</sub> and RANSAC<sub>affine</sub>. The following experiment was conducted on four well known datasets for homography estimation. All datasets include groundtruth homographies that were used to verify accuracy. First, local features were detected and matched, then each homography estimation method (RANSAC, RANSAC<sub>2pts</sub> and RANSAC<sub>affine</sub>) was applied and we declared a success if at least 80% of inliers (in consensus with the estimated homography) were in consensus with the groundtruth homography. The two steps of RANSAC (fitting and consensus) are iterated a 1000 times for each of the three variants. Therefore, the processing time spent in applying LOCATE could be compensated later on by decreasing the number of internal iterations. For equal settings, rows ‘None’ in Table 6.1 and rows ‘RANSAC’ in Table 6.2 do not correspond; this is because RANSAC USAC [RCP<sup>+</sup>13] was used in the former while baseline RANSAC in the latter.

# 7 CNN-assisted coverings in the Space of Tilts

## 7.1 Introduction

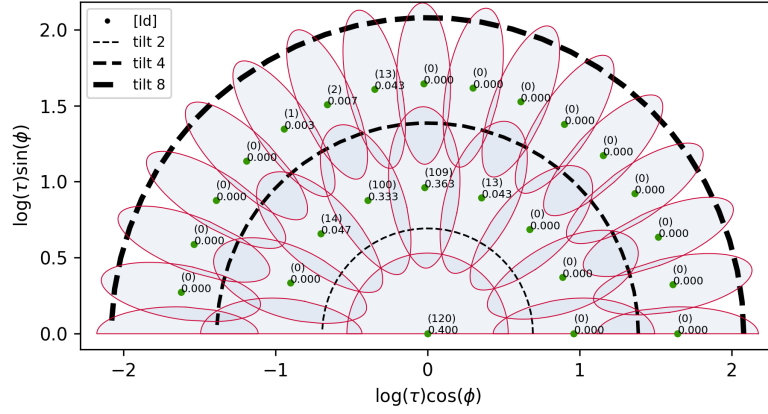
In Chapter 1, RootSIFT [AZ12] was reported to be the robustest descriptor to affine viewpoint changes (up to  $60^\circ$ ). To overcome this limitation, several *Image Matching by Affine Simulation* (IMAS) solutions have been proposed: ASIFT [YM11], FAIR-SURF [PLYP12], MODS [MMP15], Affine-AC-W in Chapter 3. Some optimal versions have been proposed in Chapter 2, including Optimal Affine-RootSIFT, which was proven to be the best choice in terms of performance. The downside of simulation-based methods is the added computations.

The recent advances in deep-learning have also contributed to the development of local descriptors. Mimicking the classic process of image matching, they learn a similarity measure between image patches [ZK15, ZL16]. In particular, affine invariance is currently being learned from data as in [MRM18] and Chapter 5. The SIFT-AID method from Chapter 5 combines SIFT keypoints with a CNN-based patch descriptor trained to capture affine invariance up to  $75^\circ$ . The Affnet method [MRM18], conceived to predict normalizing ellipse shapes for single patches based on a 3-variable parametrization, was used with HardNet [MMRM17] (a CNN-based SIIM method) to create affine invariant descriptions; its authors called this method HesAffNet. The information provided by Affnet [MRM18] can be obtained quickly but comes with a cost in precision, see Chapter 6 for more details. Still, this information concentrates in the Space of Tilts even if Affnet [MRM18] was not trained for this task. If the Affnet [MRM18] information is consistent, or at least a portion of it, then Figure 7.1 implies that Optimal Affine-RootSIFT is simulating and analysing a great number of optical views that are not needed. Figure 7.2 shows kernel density estimations in the Space of Tilts (formally introduced in Chapter 1) for query and target images in the ‘cat’ pair from the EVD [MMP15] dataset. Notice the concentration around orthogonal directions in the Space of Tilts of affine maps provided by Affnet [MRM18] from query and target images. Just by looking at those densities one can already infer that the common object to both images was seen from camera positions that differ by  $90^\circ$ .

As usual in matching methods involving normalization, each patch in HessAffnet [MRM18] is normalized to a single and possibly unprecise and/or even erroneous representation. Instead, in this chapter we propose not to rely on the precision nor on the existence of a single affine normalizing map. We prefer to compute a finite set of possible normalizing representations for each patch based on all the affine information extracted by Affnet [MRM18]. In practice, Affnet [MRM18] predictions will be used to select conve-



(a) 'cafe' target image from the EVD [MMP15] dataset.

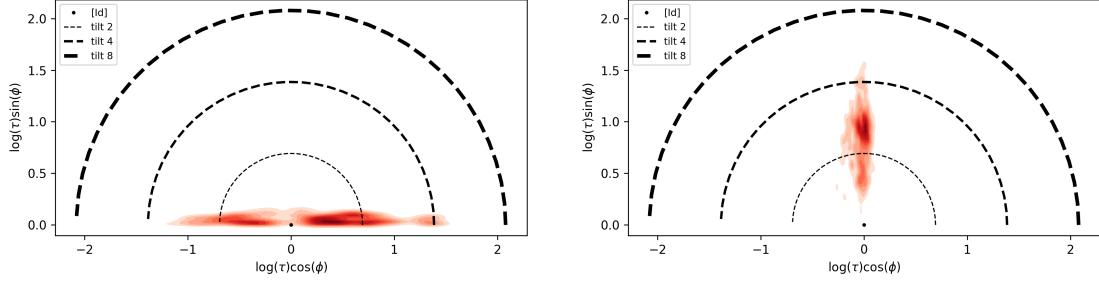


(b) The Optimal Affine-RootSIFT log 1.7-covering in the Space of Tilts. Affnet [MRM18] is applied to each patch extracted around keypoints from Figure 7.1a. These keypoints are provided by Hessian-Affine [MS04]. Centers of disks show: the number of Affnet normalizing affine maps with distances smaller than log 1.7 (top); and their corresponding percentage (bottom).

Figure 7.1: Affnet [MRM18] affine maps in the Space of Tilts.



(a) Common object to query (left) and target (right) images.



(b) Kernel density estimations of query (left) and target (right) Affnet [MRM18] affine maps.

Figure 7.2: Kernel density estimations in the Space of Tilts of affine maps extracted by Affnet [MRM18] for both images in the 'cat' pair from the EVD [MMP15] dataset.

nient affine transformations to be tested in IMAS methods. This leads to a substantial boost in IMAS speed without sacrificing performance.

The rest of this chapter is organized as follows. Section 7.2 summarizes a formal methodology for handling local viewpoint changes induced by real cameras. Two adaptive coverings based on Affnet [MRM18] are introduced in Section 7.3. They will make way for adaptive IMAS methods. The performance of the proposed methods is illustrated with experiments in Section 7.4.

## 7.2 Affine maps and the space of tilts

The Space of Tilts, denoted by  $\Omega$  and formally introduced in Chapter 1, is a quotient space where each class represents a set of affine maps with equal tilt and tilt direction (parameters  $\tau$  and  $\phi$  from Equation 1.1) and includes all possible camera spins and zooms (parameters  $\psi$  and  $\lambda$  from Equation 1.1). This space focuses on the last part  $T_\tau R_2$  of the decomposition (1.1) because it is the one that is imperfectly dealt with by most SIIM methods. Image descriptors like those proposed in the SIFT method are invariant to similarities (translations, rotations and zooms), which in terms of the camera position interpretation (see Figure 1.1) correspond to a fronto-parallel motion of the camera, a spin of the camera and to an optical zoom.

We say that two classes  $[A]$  and  $[B]$  in the Space of Tilts are equal if and only if  $T_{\tau(A)} R_{\phi(A)} = T_{\tau(B)} R_{\phi(B)}$ , where each side in this equation represents the last part of the decomposition of Equation 1.1 for  $A$  and  $B$ . As demonstrated in Proposition 1.5 from Chapter 1, the function

$$d: \begin{cases} \Omega \times \Omega & \rightarrow \mathbb{R}_+ \\ ([A], [B]) & \mapsto \log \left( \tau (BA^{-1}) \right) \end{cases} ,$$

is a metric acting on the Space of Tilts that measures the affine distortion from a fixed

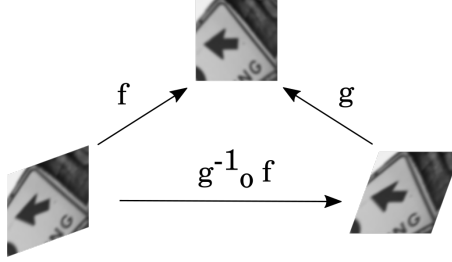


Figure 7.3: Sketch of an ideal normalization procedure.  $f, g$  two normalizing affine maps.

affine viewpoint to surrounding affine viewpoints. These distortions affect the performance of all SIIM methods (see Chapter 1) but most of them are able to successfully identify affine viewpoint distortions under  $\log 1.7$  for image sizes around  $700 \times 550$ .

In the context of image matching by affine simulation (IMAS), one crucial question to answer is: What is the best set of affine transforms to apply to each image to gain full practical affine invariance? For example, green points in Figure 7.4a represent the affine maps to be simulated on query and target images in the case of Optimal Affine-RootSIFT. Disks represent the set of affine maps distorted by no more than  $\log 1.7$  (in terms of the distance from Proposition 1.5) from the center. Notice in Figure 7.4a that a whole zone of classes with distortions up to  $\log 4\sqrt{2}$  is covered by the union of disks. This means that any distortion in that zone is reduced to less than  $\log 1.7$  from at least one of the centers. This idea of reduction is the key to the success in IMAS methods, as it ensures that any strong deformation between images can be reasonably reverted so as the matching method in question is able to cope with it.

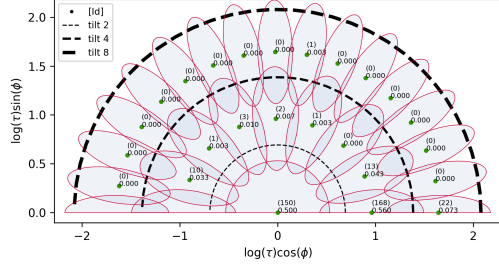
### 7.3 Adaptive coverings

The Affnet method [MRM18] is trained to predict affine-covariant region representations, where a patch is normalized before description, see Figure 7.3. The advantage of this approach is that the normalization can be obtained quickly, but at the expense of precision, see Chapter 6. On the other hand, methods like ASIFT [YM11] optically simulate affine distortions to both query and target images in order to match them. The set of simulations presented in Optimal Affine-RootSIFT from Chapter 2 correspond to an optimal  $\log 1.7$ -covering (denoted by  $\mathcal{S}_{1.7}$ ) appearing in Figure 7.4a. When Optimal Affine-RootSIFT is applied, it has been observed that most matches come from a small subset of all the affine simulations. This motivates the use of Affnet [MRM18] in order to determine an appropriate set of affine simulations to be used by IMAS methods. We call this general procedure the Adaptive IMAS method. As in the case of IMAS methods in Chapter 1, to mathematically ensure that Adaptive IMAS works one needs to:

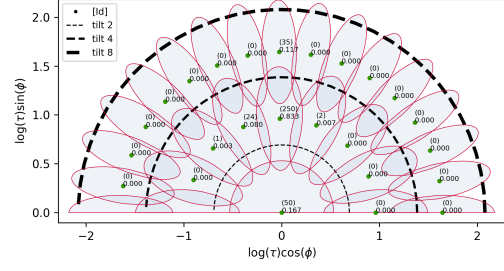
1. Dilate query and target density estimations in the Space of Tilts by a factor of  $\sqrt{r}$ , where  $r$  is the radius corresponding to the maximal viewpoint tolerance of the SIIM method (we assume  $r = 1.7$  for RootSIFT);
2. Find two sets of affine maps covering both dilated regions in step 1.

We assume that the dilation in step 1 is already taking place thanks to the already jittered information provided by Affnet [MRM18]. However, density estimations like those in Figure 7.2b are time consuming and would dramatically slow down the matching process. Instead, we propose to quickly analyze the affine information and then determine

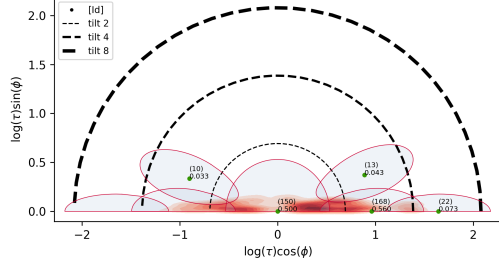




(a) Optimal Affine-RootSIFT from Chapter 2. Execution time **10.14s\***. Corresponding to 25 query and target affine simulations.



(b) Adaptive-ARootSIFT. Execution time **2.49s\***. Corresponding to 5 query and 4 target affine simulations.



(c) Greedy-ARootSIFT. Execution time **1.28s\***. Corresponding to 2 query and 2 target affine simulations.

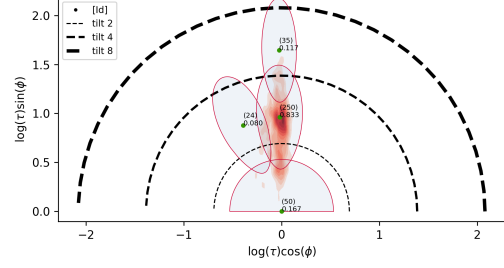


Figure 7.4: Proposed affine simulations for the ‘cat’ image pair from the EVD [MMP15] dataset. ★ OpenMP parallelization was deactivated to truly measure complexity.

two reasonable sets of affine maps (for query and target) to be simulated by an IMAS method. We now present two methodologies for building meaningful small sets of optical affine simulations for IMAS methods.

### 7.3.1 Fixed tilts selection

Here we want to determine a small (if not the smallest) subset of  $\mathcal{S}_{1.7}$  whose elements will be used to generate the simulations for the adaptive IMAS methods. This set should be such that the performance of the resulting adaptive IMAS methods is comparable to simulating the entire set  $\mathcal{S}_{1.7}$ . Algorithm 11 receives as input the information extracted by Affnet [MRM18] from a set of patches. Then, indirectly, each of these patches will vote for a transform in  $\mathcal{S}_{1.7}$  and return the set of affine maps to be simulated by an IMAS method. We call Adaptive-ARootSIFT the adaptive IMAS method whose simulations are selected by Algorithm 11 and RootSIFT is used to describe patches.

**Table 7.1** Image matching performances on three viewpoint datasets. After matching each image pair, RANSAC-USAC [RCP<sup>+</sup>13] is run 100 times to measure its probability of success in retrieving corresponding ground truth homographies. Legend: S - the number of successes (bounded by  $100 \times \boxed{\text{number}}$ ); the number of correctly matched image pairs; inl. - the average number of correct inliers; The  $\boxed{\text{numbers}}$  of image pairs in a dataset are boxed;  $N_q, N_t$  - the average number of simulated affine maps on query and target; ET - the average elapsed time in seconds. Hardware settings: (CPU) Intel i7-6700HQ 2.60GHz; (GPU) NVidia Quadro M5000M. OpenMP parallelization with 8 threads. \* Uses GPU.

Matching method	SIFT-AID dataset from Chapter 5						EVD dataset [MMP15]						OxAff dataset [MTS <sup>+</sup> 05]					
	S	$\boxed{5}$	inl.	$N_q$	$N_t$	ET	S	$\boxed{15}$	inl.	$N_q$	$N_t$	ET	S	$\boxed{40}$	inl.	$N_q$	$N_t$	ET
SIFT-AID *	500	<b>5</b>	476	1.0	1.0	4.48	100	1	159	1.0	1.0	4.32	3794	<b>38</b>	1539	1.0	1.0	7.96
RootSIFT [AZ12]	400	4	243	1.0	1.0	1.27	-	-	-	-	-	-	3900	39	1119	1.0	1.0	1.56
HesAffNet [MRM18] *	491	<b>5</b>	241	1.0	1.0	1.05	228	4	50	1.0	1.0	1.45	<b>4000</b>	<b>40</b>	576	1.0	1.0	1.20
ASIFT [YM11]	400	4	551	41.0	41.0	33.04	751	<b>9</b>	129	41.0	41.0	25.54	<b>4000</b>	<b>40</b>	5697	41.0	41.0	48.68
Optimal Affine-RootSIFT	<b>500</b>	<b>5</b>	685	25.0	25.0	5.66	<b>768</b>	<b>9</b>	186	25.0	25.0	4.96	<b>4000</b>	<b>40</b>	2794	25.0	25.0	8.12
Adaptive-ARootSIFT *	<b>500</b>	<b>5</b>	382	5.8	5.6	2.07	664	8	115	6.5	6.3	2.66	<b>4000</b>	<b>40</b>	1711	5.4	5.0	2.67
Greedy-ARootSIFT *	438	<b>5</b>	315	2.6	2.4	1.82	419	5	117	3.1	3.1	2.36	<b>4000</b>	<b>40</b>	1099	2.5	2.1	2.28

### 7.3.2 Greedy selection

We can also determine the set of simulations in a greedy iterative way until some criterion is satisfied. Algorithm 12 presents the formal procedure. Notice that  $S$  in Equation 7.2 is the current affine map in  $\tilde{\mathcal{A}}$  with more close neighbors than any other. We call Greedy-ARootSIFT the adaptive IMAS method whose simulations are selected by Algorithm 12 and RootSIFT is used to describe patches.

Figure 7.4b-7.4c illustrates the selected simulations by Adaptive-ARootSIFT and Greedy-ARootSIFT for the cat image pair in the EVD [MMP15] dataset. Notice that, when no OpenMP parallelization is used, both proposed methods run respectively 4 and 7 times faster than the Optimal Affine-RootSIFT method presented in Chapter 2. As it will be seen in our experiments, Optimal Affine-RootSIFT is still the state of the art in viewpoint performance.

## 7.4 Experiments

We now focus on the evaluation of the adaptive IMAS methods. Figures 7.5-7.6 visualize the application of the proposed methods (followed by RANSAC-USAC [RCP<sup>+</sup>13]) on two pair of images. Table 7.1 shows performances on three known datasets for homography estimation in the presence of viewpoint changes. All datasets include groundtruth homographies that were used to verify accuracy. First, correspondences from a matching method are obtained, then RANSAC-USAC [RCP<sup>+</sup>13] is applied and we declared a success if at least 80% of inliers (in consensus with the estimated homography) were in consensus with the groundtruth homography. RANSAC-USAC [RCP<sup>+</sup>13] was run 100 times to measure the probability of success in retrieving the corresponding ground truth homographies. Six metrics are reported: the number of successes; the number of correctly matched image pairs; the average number of correct inliers; the average number of affine simulations for query and target; and the average elapsed time in seconds. A perfect method would achieve the maximum number of successes in retrieving the groundtruth homography while being as fast as possible; where this maximum number of successes equals the number of images in the dataset times a hundred. A large number of matches is not an indicator of a method's good performance but can be used as tiebreaker measure

---

**Algorithm 11** Fixed Tilts Selection

---

**input:** $\mathcal{A}$  - Set of normalizing affine maps provided by Affnet [MRM18] from all patches of an image.**parameters:** $r$  - Tilt radius (default to 1.7). $S_r$  - Set of optimal affine simulations (default to  $S_{1.7}$ ). $\alpha$  - Cover threshold (default to 0.01).**start:** $\mathcal{S}_{FT} = \emptyset.$ 

▷ initialization

**foreach**  $S \in \mathcal{S}_r$  **do**

$$p = \frac{\sum_{A \in \mathcal{A}} \mathbb{1}_{d([A], [S]) \leq \log r}}{|\mathcal{A}|}. \quad (7.1)$$

**if**  $p \geq \alpha$  **then** $\mathcal{S}_{FT} = \mathcal{S}_{FT} \cup \{S\}.$ **return**  $\mathcal{S}_{FT}$ 

---

---

**Algorithm 12** Greedy Selection

---

**input:** $\mathcal{A}$  - Set of normalizing affine maps provided by Affnet [MRM18] from all patches of an image.**parameters:** $r$  - Tilt radius (default to 1.7). $\alpha$  - Cover threshold (default to 0.05).**start:** $\tilde{\mathcal{A}} = \mathcal{A}, \mathcal{S}_G = \emptyset.$ 

▷ initialization

**while**  $|\tilde{\mathcal{A}}| \geq \alpha |\mathcal{A}|$  **do**

$$S = \arg \max_{S \in \tilde{\mathcal{A}}} \sum_{A \in \tilde{\mathcal{A}}} \mathbb{1}_{d([A], [S]) \leq \log r}. \quad (7.2)$$

 $\mathcal{S}_G = \mathcal{S}_G \cup \{S\}.$  $\tilde{\mathcal{A}} = \tilde{\mathcal{A}} \setminus \{[A] \in \Omega \mid d([A], [S]) \leq \log r\}.$ **return**  $\mathcal{S}_G$ 

---

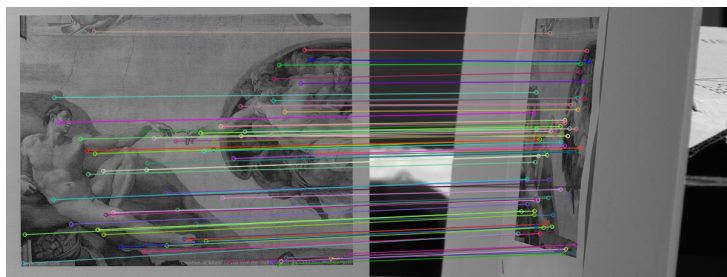
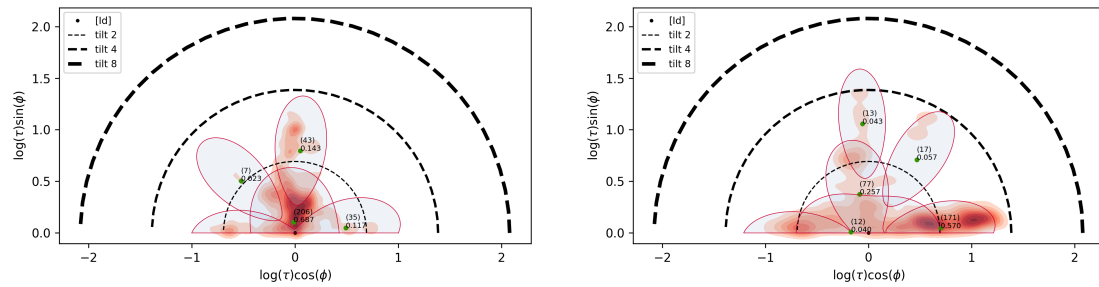
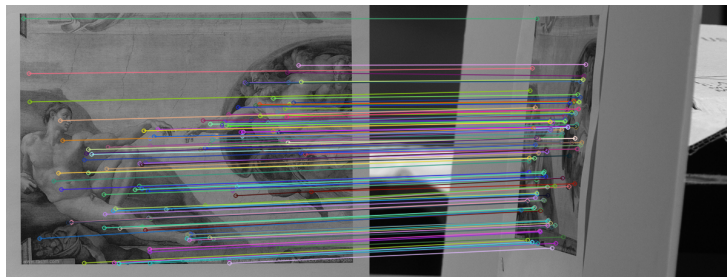
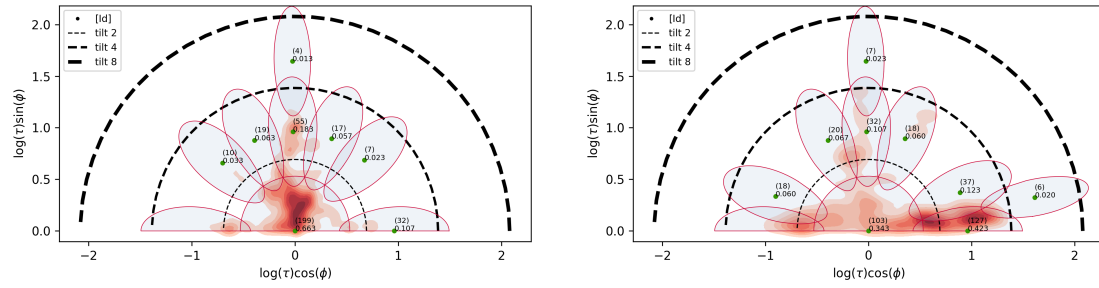
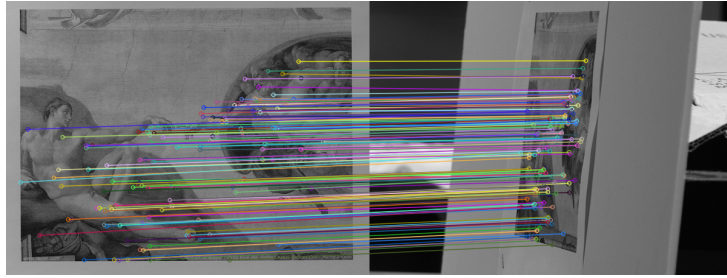
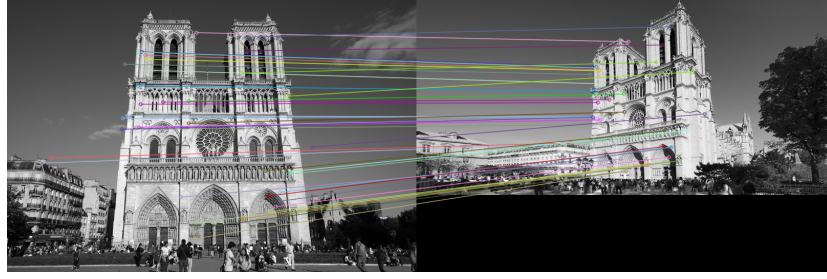
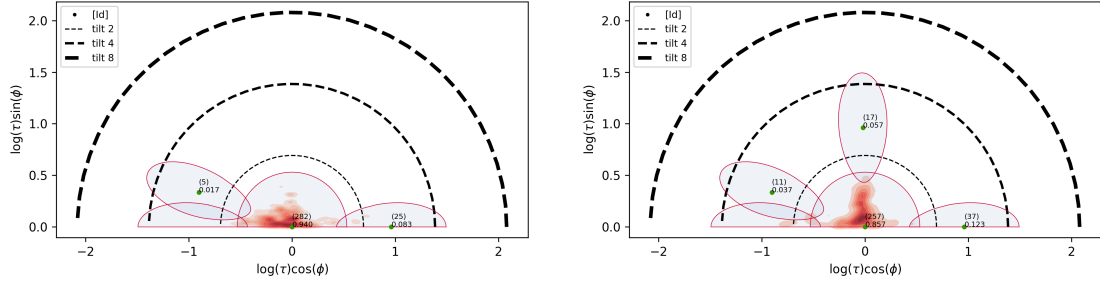


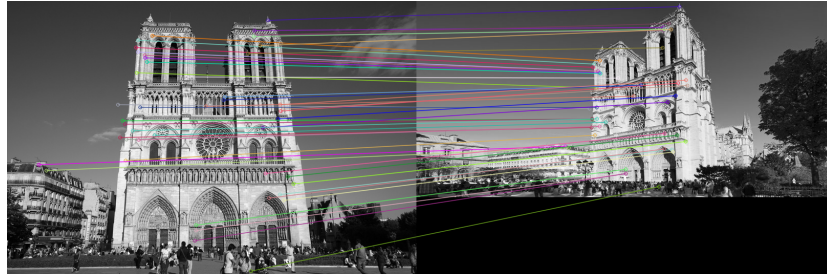
Figure 7.5: Proposed affine simulations for the ‘adam’ image pair from the EVD [MMP15] dataset.



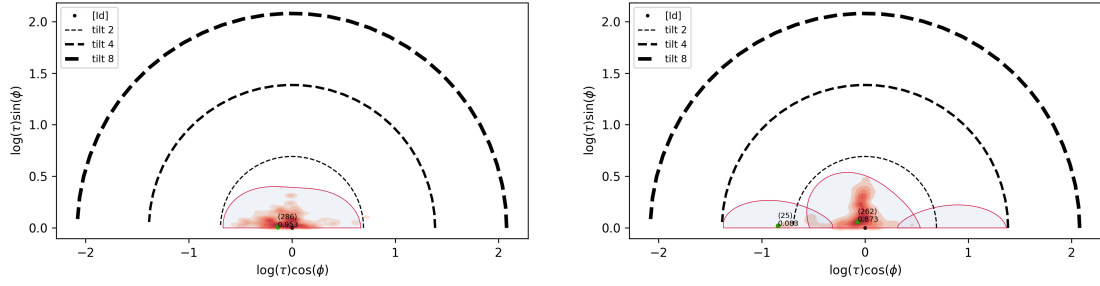
(a) Optimal Affine-RootSIFT results for the 'notredame' image pair.



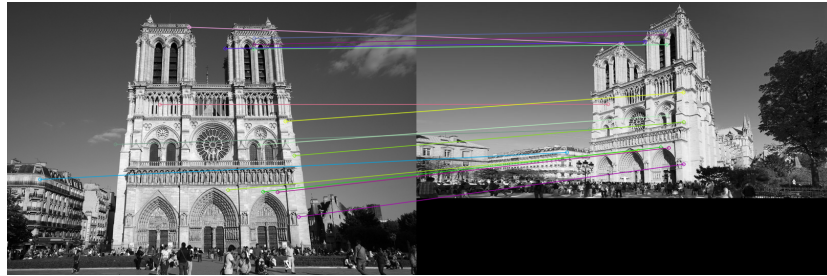
(b) Results of Algorithm 11 for query and target images from the 'notredame' image pair.



(c) Adaptive-ARootSIFT results for the 'notredame' image pair.



(d) Results of Algorithm 12 for query and target images from the 'notredame' image pair.



(e) Greedy-ARootSIFT results for the 'notredame' image pair.

Figure 7.6: Proposed affine simulations for the 'notredame' image pair from the Chapter 5 dataset.

if two methods are equally good in identifying geometric models.

As was been pointed out in Chapter 5, IMAS methods benefit from lots of keypoints that come exclusively from simulated versions of the input images. Indeed, SIIM detectors themselves are not affine invariant. Therefore, the more affine simulations in an IMAS method, the larger amount of matches it will possibly recognize. Notice in Table 7.1, for the OxAff dataset [MTS<sup>+</sup>05], that Optimal Affine-RootSIFT from Chapter 2 has far fewer matches on average than ASIFT [YM11]. However, as previously stated, the number of matches might be misleading about the method’s true performance. Table 7.1 points out that Optimal Affine-RootSIFT from Chapter 2 performs better than ASIFT [YM11] in two datasets; indeed, the former method has more successes in retrieving groundtruth homographies (i.e. larger probability of success) with even one more identified pairs of images in the SIFT-AID dataset from Chapter 5. With this in mind, we can declare Optimal Affine-RootSIFT to be state of the art in viewpoint invariant image matching. On the other hand, execution times of Optimal Affine-RootSIFT from Chapter 2 are higher than non-simulating methods but still considerably faster than ASIFT [YM11].

Table 7.1 shows that adaptive IMAS methods provide a good compromise between performance and speed. Adaptive-ARootSIFT attains the same level of performance of Optimal Affine-RootSIFT (best in all three datasets) in successfully identifying groundtruth homographies while reducing by half the average computing time with best case scenario reduced by four. Even if not as fast as Affnet [MRM18], Adaptive-ARootSIFT provides a remarkable boost in successes and identified image pairs with respect to the former method, highlighted in the EVD [MMP15] dataset. HessAffnet [MRM18] was forced to detect 2000 keypoints and, as in [MRM18], incorporates the HardNet [MMRM17] descriptor. The average number of simulations in Greedy-ARootSIFT has halved with respect to Adaptive-ARootSIFT. This last fact is not quite perceived in execution times of Table 7.1 due to parallelism but is best appreciated in Figure 7.4 where parallelism was deactivated.



## 8 Conclusion

In Chapter 5 we proposed a CNN image patch descriptor capturing affine invariance without the necessity of using viewpoint simulations nor affine normalisation. In our experiment, that uses pairs of images from Figure 5.6, the SIFT-AID method attains a performance comparable to Optimal Affine-RootSIFT. Yet, AID was trained with a very simplistic data generation scheme in order to be invariant to contrast and affine viewpoint changes. Even more, pairs of AID descriptors are classified based on a Bayes predictor whose deciding threshold was fixed for synthetic data; whereas RootSIFT uses the well established second nearest neighbor criterion. Lowe’s criterion has been proved to enhance distinctiveness, but the Bayes predictor is compatible with repetitive structures.

An ideal matching method returns a big set of matches with the highest ratio of true positives. AID’s distinctiveness have been assessed by Figure 5.5 and Table 5.1. Unfortunately, most of the missing matches of the SIFT-AID method in Table 5.1 are due to SIFT’s keypoint detection step failures; more work is needed to improve this step. Indeed, provided that proper keypoints had been correctly spotted in the first stages of the SIFT detector, AID would have been sufficient to identify almost all Affine-RootSIFT matches. Figure 8.1 shows missed and retrieved matches from AID measurements with respect to all correctly identified matches from Optimal Affine-RootSIFT. AID matches are plotted with respect to zoom and viewpoint angle differences between patches. Both quantities are computed from local approximating affine maps (first order Taylor approximations) of ground truth homographies. Notice that most of the missing matches between AID descriptors involve viewpoint angles close to 75 degrees, the maximal viewpoint angle present when training the BigAID network. Most failures coming from the graffiti pair.

If improved, the still evolving ideas described in Chapter 5 could lead to more robust AID descriptions. Some key thoughts to enhance AID related matching methods are:

- *A more realistic data generation scheme for retraining AID.* For example, in order to improve distinctiveness, we can select the negative patch from the anchor image itself. Random noise with random standard deviation could also be added to all three patches before feeding them to the hinge loss. Simulating real occlusions is another important feature for intra-class patch generation.
- *Optimal patch-size.* An input patch-size attaining a good trade-off between locality and description is desired.
- *Network architecture.* Optimize the descriptor network architecture to improve performance and speed.
- *A geometric model estimator robust to multiple-to-multiple matches.* A model fitting procedure capable of dealing with multiple matches caused by, for example, repetitive structures.



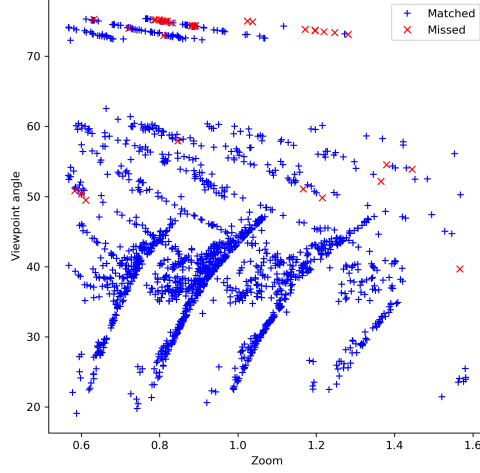


Figure 8.1: AID classification on oracle SIFT keypoints corresponding to true Optimal Affine-RootSIFT matches. Each of these SIFT keypoints is replacing an IMAS-like keypoint so as similarity deformations around true Optimal Affine-RootSIFT matches are minimal. Notice that IMAS keypoints also reduce tilt deformations, whereas SIFT keypoints do not. All pair of images from Figure 5.6 were used.

- *Improved affine viewpoint invariance.* Further extend the viewpoint robustness of AID by combining it with affine simulations techniques similar to those in Chapter 1 and/or affine patch normalization like Affnet [MRM18].

Some of these ideas we have already started to explore. First,  $\text{RANSAC}_{\text{affine}}$  from Chapter 6 is used in Table 8.1 to impose geometry consistency when estimating an homography from AID matches; avoiding in this way, the need of Lowe’s second nearest neighbor criterion. Second, we improve still more AID’s affine invariant descriptions by combining it in Table 8.1 with the Hessian-Affine [MS04] detector plus the Affnet [MRM18] affine patch normalizer. Third, a new data generation procedure is proposed in Figure 8.2 to generate more realistic ROC curves. Finally, the dimension of the AID descriptor can be further reduced, as seen in Figure 8.3.

Table 7.1 shows a rather poor performance of the SIFT-AID method. We want to measure the extent to which it is AID’s fault and the gap to improve. Figure 8.2 shows ROC curves for four state-of-the-art descriptions: AID, Affnet [MRM18]+Hardnet [MMRM17], Hardnet [MMRM17] and RootSIFT [AZ12]. AID has the best score among descriptors, only beaten by Affnet [MRM18]+Hardnet [MMRM17]. In order to measure the state of the art descriptions’ performances we test them within a unique detector, so as all of those methods receive the same set of patches to describe. We choose the Hessian-Affine [MS04] detector and keep the best 500 keypoints for each image. Table 8.1 shows similar performances between AID equipped with  $\text{RANSAC}_{\text{affine}}$  from Chapter 6 and HessAffnet [MRM18]; the later method appearing in Table 8.1 as HessAffnetHardnet + USAC. The combination of AID and Affnet [MRM18] results in an improvement over AID when  $\text{RANSAC-USAC}$  [RCP<sup>+</sup>13] is used; however, their performance look similar when followed by  $\text{RANSAC}_{\text{affine}}$ . This proves that AID does provide state-of-the-art affine invariant descriptions while allowing repetitive structures to be matched. The main drawback of AID is its slow keypoint descriptions step. We point out that the Affnet [MRM18] network receives as input  $32 \times 32$  patches whereas the BigAID descriptor receives  $60 \times 60$

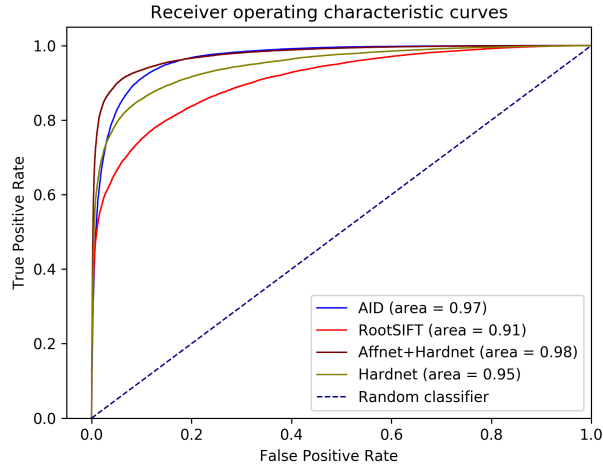


Figure 8.2: ROC curves and their corresponding AUC for AID, Affnet [MRM18]+Hardnet [MMRM17], Hardnet [MMRM17] and RootSIFT [AZ12]. 60000 pair of generated patches were used; half intra-class (positives), half extra-class (negatives). The triple patch generation of Chapter 5 was enriched by two main additions: first, the negative patch belongs to the anchor image; second, random noise with random standard deviation is added to all three patches.

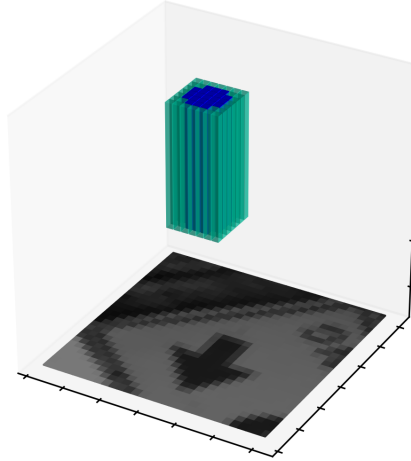


Figure 8.3: The AID21 descriptor shares  $21 \times 128$  dimensions (depicted in blue) out of  $49 \times 128$  from the AID descriptor. The AID descriptor corresponds to the whole structure in blue and green. All dimensions in green are describing patch zones near the borders, whereas all those in blue correspond to patch zone's descriptions around the center.

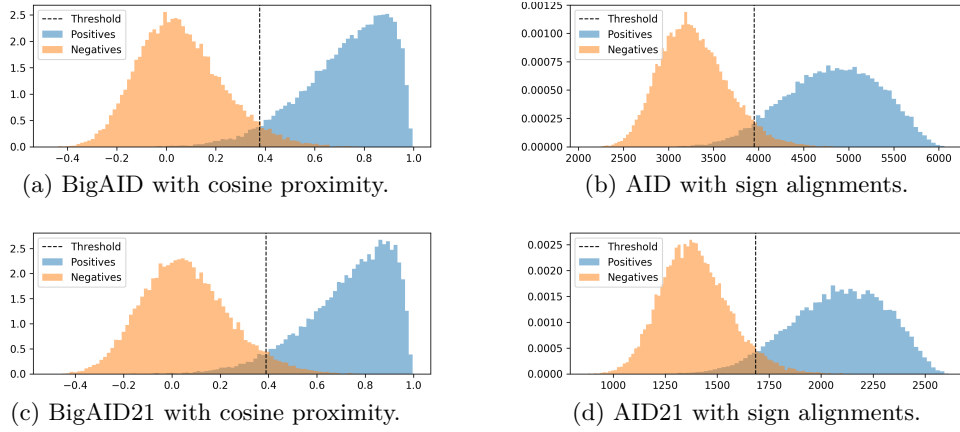


Figure 8.4: Positive and negative density estimation on measurements. For that,  $3 \cdot 10^5$  random intra and extra class pairs were used. The vertical line depicts the threshold minimizing both error probabilities: false negatives and false positives.

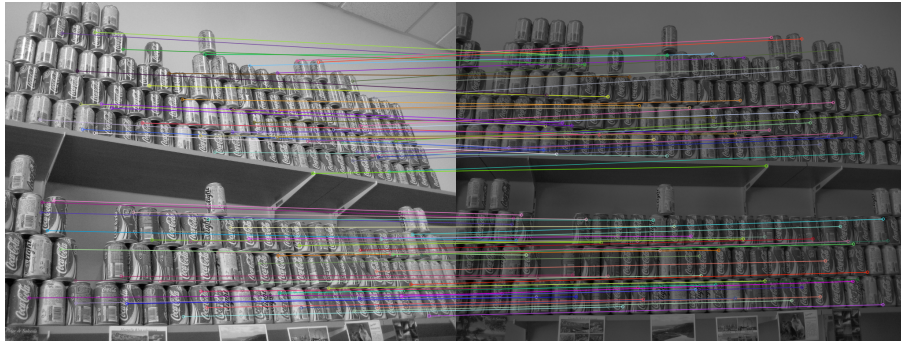
patches. Even more, the BigAID network doubles the number of layers and filters with respect to Hardnet and Affnet networks. Clearly, the BigAID network is oversized and needs to be optimized for faster descriptions. On the other hand, the matching of AID descriptors is already done fast. AID descriptions can be further reduced into AID21 descriptions, resulting in the fastest brute force matching step from all methods described in this thesis. The AID21 descriptor accounts for a reduction from 6272 bits (196 floats) to 2688 bits (84 floats), whereas SIFT descriptions need 128 floats. This passage from AID to AID21 is done without sacrificing distinctiveness nor performance (see Figure 8.4 and Table 8.1), which highlights over-descriptions from the BigAID network. All this suggests that a shrinkage of the BigAID network is indeed highly possible. Even more, the AID21 descriptor seems to require less help from Affnet [MRM18] than AID, for both RANSAC<sub>affine</sub> and RANSAC-USAC [RCP<sup>+</sup>13]. This improvement, observed in Table 8.1, might be due to the more local descriptions provided by AID21.

Chapter 6 proposes a CNN based method to locally estimate affine maps between images. These local affine maps provide us with tangent planes of the global transformation from the query image to the target image. Under reasonable assumptions, tangent planes can determine the global transformation. Figure 8.5 proves that even a simple local affine reconstruction with LOCATE already provides a fair enough reconstruction. Blurry details in the reconstructed image were expected even for a perfect method. Indeed, there exist two main reasons for that:

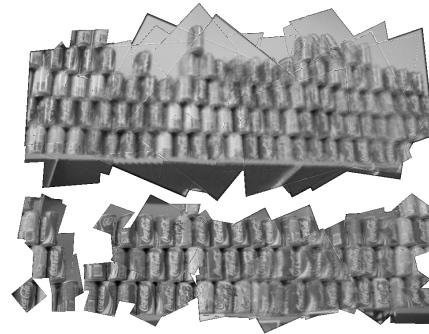
1. The reconstruction is done with first order approximations. Therefore, there is always an approximation error that causes blur when overlapping patches are averaged.
2. Some query patches belong to higher order scales in the Gaussian pyramid. They are already blurry by definition.

Our experiments from Chapter 6 show that the LOCATE method provides accurate first-order approximations of local geometry. This information proved to be valuable for two applications:

- Guided matching of SIFT keypoints with precise locations, orientations and scales.



(a) Initial correspondences.



(b) LOCATE affine reconstruction of target from query.

Figure 8.5: LOCATE’s local to global reconstruction. First, some correspondences are provided by some matching method, as in Figure 8.5a. Then we compute local approximating affine maps with LOCATE around each of these correspondences and use them to transform query patches into the target image plane, see Figure 8.5b. Each pixel intensity is averaged with the information of all patches enclosing it.

**Table 8.1** Image matching performances on three viewpoint datasets. After matching each image pair, a RANSAC is run 100 times to measure its probability of success in retrieving corresponding ground truth homographies. Legend: S - the number of successes (bounded by  $100 \times \boxed{\text{number}}$ ); the number of correctly matched image pairs; inl. - the average number of correct inliers; AvE - the average pixel error; R - the ratio of inliers/total. The  $\boxed{\text{numbers}}$  of image pairs in a dataset are boxed. RANSAC and RANSAC<sub>affine</sub> are both from Chapter 6 and used with 10000 internal iterations; \* and \*\* mean the usage of local affine maps provided by the HessAff detector and the Affnet normaliser, respectively. USAC stands for RANSAC-USAC [RCP<sup>+</sup>13] with standard settings.

Matching method (+ RANSAC version)	SIFT-AID dataset from Chapter 5					EVD dataset [MMP15]					OxAff Viewpoint dataset [MTS <sup>+</sup> 05]				
	S	$\boxed{5}$	inl.	AvE	R	S	$\boxed{15}$	inl.	AvE	R	S	$\boxed{10}$	inl.	AvE	R
HessAffAID (RANSAC <sub>affine</sub> *)	414	5	97	5.8	0.21	195	3	18	8.5	0.04	991	10	116	3.6	0.25
HessAffnetAID (RANSAC <sub>affine</sub> **)	422	5	101	6.0	0.22	200	2	23	6.3	0.05	1000	10	112	3.8	0.24
HessAffAID (USAC)	357	4	119	5.5	0.25	100	1	44	6.0	0.09	800	8	146	2.7	0.32
HessAffnetAID (USAC)	300	3	136	4.3	0.30	102	2	52	7.1	0.11	803	9	143	2.9	0.31
HessAffAID21 (RANSAC <sub>affine</sub> *)	466	5	82	6.0	0.18	197	2	17	7.3	0.04	999	10	110	3.8	0.24
HessAffnetAID21 (RANSAC <sub>affine</sub> **)	430	5	89	6.1	0.20	198	2	16	7.0	0.03	998	10	106	3.7	0.23
HessAffAID21 (USAC)	350	4	111	5.4	0.24	25	1	37	6.3	0.08	795	8	141	2.7	0.31
HessAffnetAID21 (USAC)	300	3	126	4.1	0.28	99	1	46	7.5	0.10	800	8	136	2.7	0.30
HessAffnetHardnet (USAC)	446	5	78	4.0	0.62	200	2	13	4.1	0.49	1000	10	109	2.5	0.84
HessAffnetHardnet (RANSAC)	400	4	87	4.0	0.67	200	2	14	4.4	0.52	1000	10	110	2.6	0.85

- Homography estimation, for which we presented a RANSAC version that systematically improved results in four well known datasets [ZR11, MMP15, MTS<sup>+</sup>05, HS12].

The proposed method is generic and its applications to stereo matching without any global geometry assumption will be explored in future work. Training LOCATE to handle occlusions is crucial for real-life applications, and will also be the focus of future work.

In Chapter 7 we show that Image matching by affine simulation (IMAS) methods are still the state of the art in matching images involving strong viewpoint differences. Even when compared to recent advances incorporating neural networks like [MRM18] or SIFT-AID from Chapter 5, IMAS methods do perform better. We observe that the information provided by AffNet [MRM18] is valuable in determining convenient simulations to be used in IMAS methods. The resulting adaptive IMAS methods yield a substantial acceleration with respect to classic IMAS methods without sacrificing performance. Also, Equation 7.1 provides a natural order to simulations appearing in Optimal Affine-RootSIFT which will be used in future work to create IMAS methods that gradually incorporate simulations on demand and stop as soon as a significant geometric model (i.g., homography) has been identified.

# Appendices





# A Proof of Theorem 1.1

In this chapter we provide the proof of Theorem 1.1. For the sake of clearness, we state again Theorem 1.1.

**Theorem.** *Given an element of the space of tilts in canonical form  $[T_t R(\phi_1)]$ , the disk  $\mathcal{B}([T_t R(\phi_1)], r)$  in the space of tilts centered at this element and with radius  $r$  corresponds to the following set*

$$\left\{ [T_s R(\phi_2)] \mid G(t, s, \phi_1, \phi_2) \leq \frac{e^{2r} + 1}{2e^r} \right\}$$

where

$$G(t, s, \phi_1, \phi_2) = \left( \frac{\frac{t}{s} + \frac{s}{t}}{2} \right) \cos^2(\phi_1 - \phi_2) + \left( \frac{\frac{1}{st} + st}{2} \right) \sin^2(\phi_1 - \phi_2).$$

*Proof.* By proposition 1.4 we know that

$$\tau(BA^{-1}) = \tau(i([B])i([A])^{-1})$$

where  $i$  is the injection in Definition 1.5. Thus, without loss of generality, we focus in computing the absolute tilt of

$$\begin{aligned} C &= T_t R_2 Q_2^{-1} T_s^{-1} \\ &= T_t R(\phi) T_s^{-1} \end{aligned}$$

where  $R(\phi) = R_2 Q_2^{-1}$ . Proposition 1.2 states that the ratio between the singular values of  $C$  can be used to compute its absolute tilt.

## Trace and determinant

First, we start by computing the trace and determinant of

$$C^* C = T_s^{-1} R(\phi)^{-1} T_t T_t R(\phi) T_s^{-1},$$

which are clearly

$$\det(C^* C) = \frac{t^2}{s^2}$$

and

$$\text{Tr}(C^* C) = \left( \frac{t^2}{s^2} + 1 \right) \cos^2 \phi + \left( \frac{1}{s^2} + t^2 \right) \sin^2 \phi.$$

### The eigenvalues of $C^*C$

Let  $H = \begin{pmatrix} a & c \\ c & b \end{pmatrix} = C^*C$  and  $\lambda_+, \lambda_-$  being the biggest and smallest eigenvalues of  $C^*C$  respectively. It is well known that

$$\begin{aligned} \text{Tr}(H) &= \lambda_+ + \lambda_- \\ \det(H) &= \lambda_+ \lambda_- \end{aligned}$$

and even more that both  $\text{Tr}$  and  $\det$  also appear in the characteristic polynomial

$$\begin{aligned} |C^*C - \lambda Id| &= \lambda^2 - \lambda(a + b) + (ab - c^2) \\ &= \lambda^2 - \lambda \text{Tr} H + \det H. \end{aligned}$$

On the other hand, the eigenvalues of a symmetric positive definite matrix are in  $\mathbb{R}$ , which implies that  $\sqrt{(\text{Tr} H)^2 - 4 \det H} \geq 0$ , and so one can write

$$\begin{aligned} \lambda_- &= \frac{\text{Tr}(H) - \sqrt{(\text{Tr} H)^2 - 4 \det H}}{2}, \\ \lambda_+ &= \frac{\text{Tr}(H) + \sqrt{(\text{Tr} H)^2 - 4 \det H}}{2}. \end{aligned}$$

Now, after some computations, the ratio between the biggest and smallest eigenvalues is

$$\begin{aligned} \frac{\lambda_+}{\lambda_-} &= \frac{\left( \frac{\text{Tr} H}{2} + \frac{\sqrt{(\text{Tr} H)^2 - 4 \det H}}{2} \right)^2}{\det H} \\ &= \frac{s^2}{t^2} \left( \frac{g}{2} + \frac{\sqrt{g^2 - 4 \frac{t^2}{s^2}}}{2} \right)^2 \end{aligned} \tag{A.1}$$

where  $g$  denotes the function

$$\begin{aligned} g(t, s, \phi) &:= \text{Tr}(C^*C) \\ &= \left( \frac{t^2}{s^2} + 1 \right) \cos^2 \phi + \left( \frac{1}{s^2} + t^2 \right) \sin^2 \phi. \end{aligned}$$

### Computing $\tau(C)$

Proposition 1.2 tells that the absolute tilt of  $C$  is

$$\begin{aligned} \tau(C) &= \sqrt{\frac{\lambda_+}{\lambda_-}} \\ &= \frac{s}{t} \left( \frac{g}{2} + \frac{\sqrt{g^2 - 4 \frac{t^2}{s^2}}}{2} \right) \\ &= \frac{s}{t} \frac{g}{2} + \sqrt{\left( \frac{s}{t} \frac{g}{2} \right)^2 - 1} \\ &= G(s, t, \phi) + \sqrt{(G(s, t, \phi))^2 - 1} \end{aligned}$$

where

$$G(s, t, \phi) = \frac{s}{t} \frac{g(s, t, \phi)}{2}.$$

### Disks in the space of tilts

Let  $\mathbf{A} := [T_t R_2] \in \Omega$  be fixed and let us find conditions on  $\mathbf{B} := [T_s Q_2] \in \Omega$  to satisfy

$$\mathbf{B} \in \mathcal{B}(\mathbf{A}, \log r)$$

which are clearly

$$\begin{aligned} d(\mathbf{A}, \mathbf{B}) &= \log \tau \left( i(\mathbf{A}) i(\mathbf{B})^{-1} \right) \leq \log r \\ &\Downarrow \\ \tau \left( i(\mathbf{A}) i(\mathbf{B})^{-1} \right) &\leq r \end{aligned}$$

where  $i$  is the injection in Definition 1.5. Thus, just by applying the above to  $C := i(\mathbf{A}) i(\mathbf{B})^{-1}$  we obtained

$$\begin{aligned} G(s, t, \phi) + \sqrt{(G(s, t, \phi))^2 - 1} &= \tau(AB^{-1}) \\ &\leq r \end{aligned}$$

where  $R(\phi) = R_2 Q_2^{-1}$ . So

$$\begin{aligned} \sqrt{G^2 - 1} &\leq r - G \\ &\Downarrow \\ G^2 - 1 &\leq r^2 - 2rG + G^2 \\ &\Downarrow \\ G &\leq \frac{r^2 + 1}{2r}. \end{aligned}$$

□



# Bibliography

- [AAC<sup>+</sup>06] A Agarwala, M Agrawala, M Cohen, D Salesin, and R Szeliski. Photographing long scenes with multi-viewpoint panoramas. *International Conference on Computer Graphics and Interactive Techniques*, pages 853–861, 2006.
- [ABD12] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. KAZE features. In *Lecture Notes in Computer Science*, volume 7577 LNCS, pages 214–227, 2012.
- [AFS<sup>+</sup>11] Sameer Agarwal, Yasutaka Furukawa, Noah Snavely, Ian Simon, Brian Curless, Steven M Seitz, and Richard Szeliski. Building rome in a day. *Communications of the ACM*, 54(10):105–112, 2011.
- [ANB13] Pablo Fernández Alcantarilla, Jesús Nuevo, and Adrien Bartoli. Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces. *British Machine Vision Conference*, pages 13.1–13.11, 2013.
- [ATRB95] A P Ashbrook, N A Thacker, P I Rockett, and C I Brown. Robust recognition of scaled shapes using pairwise geometric histograms. *BMVC*, pages 503–512, 1995.
- [AZ12] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.
- [Bau00] A. Baumberg. Reliable feature matching across widely separated views. *CVPR*, 1:774–781, 2000.
- [BH16] Daniel Barath and Levente Hajder. Novel ways to estimate homography from local affine transformations. 2016.
- [BL03] M Brown and D Lowe. Recognising panoramas. In *Proc. the 9th Int. Conf. Computer Vision, October*, pages 1218–1225, 2003.
- [Blo92] J. Blom. Topological and Geometrical Aspects of Image Structure. *University of Utrecht*, 1992.
- [BMP02] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *TPAMI*, 24(4):509–522, 2002.
- [BS11] Matthew Brown and Sabine Süsstrunk. Multi-spectral SIFT for scene category recognition. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 177–184. IEEE, 2011.

- [BSBB06] M. Bennewitz, C. Stachniss, W. Burgard, and S. Behnke. Metric Localization with Scale-Invariant Visual Features Using a Single Perspective Camera. *European Robotics Symposium*, 2006.
- [BTV06] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *ECCV*, 1:404–417, 2006.
- [Cha05] E. Y. Chang. EXTENT: fusing context, content, and semantic ontology for photo annotation. *Proceedings of the 2nd international workshop on Computer vision meets databases*, pages 5–11, 2005.
- [CLM<sup>+</sup>08] F. Cao, J.-L. Lisani, J.-M. Morel, P. Musé, and F. Sur. *A Theory of Shape Identification*. Springer Verlag, 2008.
- [CLSF10] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *Lecture Notes in Computer Science*, volume 6314 LNCS, pages 778–792, 2010.
- [CMK03] O. Chum, J. Matas, and J. Kittler. Locally Optimized RANSAC. *Proceedings of the DAGM*, 2781:236–243, 2003. [https://doi.org/10.1007/978-3-540-45243-0\\_31](https://doi.org/10.1007/978-3-540-45243-0_31).
- [CPV15] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, 2015.
- [DMM08] A. Desolneux, L. Moisan, and J.-M. Morel. *From Gestalt Theory to Image Analysis*. Springer, 2008.
- [DMPC10] Petr Douthek, Jiri Matas, Michal Perdoch, and Ondrej Chum. Image matching and retrieval by repetitive patterns. In *20th International Conference on Pattern Recognition (ICPR)*, pages 3195–3198. IEEE, 2010.
- [DMR16] D. DeTone, T. Malisiewicz, and A. Rabinovich. Deep image homography estimation. *arXiv preprint arXiv:1606.03798*, 2016.
- [Fau93] O Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.
- [FB81] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
- [FBA<sup>+</sup>06] Q Fan, K Barnard, A Amir, A Efrat, and M Lin. Matching slides to presentation videos using SIFT and scene background matching. *Proceedings of the 8th ACM international workshop on Multimedia information retrieval*, pages 239–248, 2006.
- [Fév07] L Février. A wide-baseline matching library for Zeno. *Internship report*, [www.di.ens.fr/~fevrier/papers/2007-InternsipReportILM.pdf](http://www.di.ens.fr/~fevrier/papers/2007-InternsipReportILM.pdf), 2007.
- [FH15] Erez Farhan and Rami Hagege. Geometric expansion for local feature analysis and matching. *SIAM Journal on Imaging Sciences*, 8(4):2771–2813, 2015.

- [FMH17] Erez Farhan, Elad Meir, and Rami Hagege. Local Region Expansion: a Method for Analyzing and Refining Image Matches. *Image Processing On Line*, 7:386–398, 2017.
- [FS07] Jun Jie Foo and Ranjan Sinha. Pruning SIFT for scalable near-duplicate image matching. In *ADC '07: Proceedings of the eighteenth conference on Australasian database*, pages 63–71, Darlinghurst, Australia, Australia, 2007. Australian Computer Society, Inc.
- [FSKP] G Fritz, C Seifert, M Kumar, and L Paletta. Building detection from mobile imagery using informative SIFT descriptors. *Lecture Notes in Computer Science*, pages 629–638.
- [GGQY07] A. Gordon, G. Glazko, X. Qiu, and A. Yakovlev. Control of the mean number of false discoveries, bonferroni and stability of multiple testing. *Ann. Appl. Stat.*, 1:179–190, 2007.
- [GL06] I Gordon and D G Lowe. What and Where: 3D Object Recognition with Accurate Pose. *Lecture Notes in Computer Science*, 4170:67, 2006.
- [GLGP13] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2916–2929, 2013.
- [GvGP15] R. Grompone von Gioi and V. Pătrăucean. A contrario patch matching, with an application to keypoint matches validation. In *ICIP*, pages 946–950. 2015.
- [GZS11] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 963–968. Ieee, 2011.
- [HL04] J S Hare and P H Lewis. Salient regions for query by image content. *Image and Video Retrieval: Third International Conference, CIVR*, pages 317–325, 2004.
- [HS88] C Harris and M Stephens. A combined corner and edge detector. *Alvey Vision Conference*, 15:50, 1988.
- [HS12] D. C. Hauagge and N. Snavely. Image matching using local symmetry features. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 206–213. IEEE, 2012.
- [HZ03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [Iij71] T. Iijima. Basic equation of figure and and observational transformation. *Systems, Computers, Controls*, 2(4):70–77, 1971.
- [Kar16] Maxim Karpushin. *Local features for RGBD image matching under view-point changes*. PhD thesis, 2016.



- [KRTA13] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-Match: Fast Affine Template Matching. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1940–1947. IEEE, 2013.
- [KS04] Y Ke and R Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. *CVPR*, 2:506–513, 2004.
- [KSS07] A. Kelman, M. Sofka, and C.V. Stewart. Keypoint descriptors for matching across multiple image modalities and non-linear intensity variations. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007. <https://doi.org/10.1109/CVPR.2007.383426>.
- [KZB04] T Kadir, A Zisserman, and M Brady. An Affine Invariant Salient Region Detector. In *ECCV*, pages 228–241, 2004.
- [LÁJA06] Herwig Lejsek, Fridrik H Ásmundsson, Björn Thór Jónsson, and Laurent Amsaleg. Scalability of local image descriptors: a comparative study. In *MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia*, pages 589–598, New York, NY, USA, 2006. ACM.
- [LCC06] B N Lee, W Y Chen, and E Y Chang. Fotofiti: web service for photo management. *Proceedings of the 14th annual ACM international conference on Multimedia*, pages 485–486, 2006.
- [LCS11] Stefan Leutenegger, Margarita Chli, and Roland Y. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2548–2555, 2011.
- [LE06] G Loy and J O Eklundh. Detecting symmetry and symmetric constellations of features. *Proceedings of ECCV*, 2:508–521, 2006.
- [LG94] T. Lindeberg and J. Garding. Shape-adapted smoothing in estimation of 3-D depth cues from affine distortions of local 2-D brightness structure. *ECCV*, pages 389–400, 1994.
- [LH16] Gil Levi and Tal Hassner. LATCH: Learned arrangements of three patch codes. In *2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016*, 2016.
- [Lin93] T. Lindeberg. *Scale-Space Theory in Computer Vision*. Royal Institute of Technology, Stockholm, Sweden, 1993.
- [Lin95] T. Lindeberg. Direct estimation of affine image deformations using visual front-end operations with automatic scale selection. In *Computer Vision, 1995. Proceedings., Fifth International Conference on*, pages 134–141. IEEE, 1995.
- [Lin11] T. Lindeberg. Generalized gaussian scale-space axiomatics comprising linear scale-space, affine scale-space and spatio-temporal scale-space. *Journal of Mathematical Imaging and Vision*, 40(1):36–81, 2011.

- [Lin13a] T. Lindeberg. Invariance of visual operations at the level of receptive fields. *BMC Neuroscience*, 14(1):P242, 2013.
- [Lin13b] T. Lindeberg. Scale selection properties of generalized scale-space interest point detectors. *JMIV*, 46(2):177–210, 2013.
- [LMB<sup>+</sup>14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.
- [Low85] D. Lowe. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers, 1985.
- [Low04] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [LYT11] Ce Liu, Jenny Yuen, and Antonio Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):978–994, 2011.
- [MCUP04] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *IVC*, 22(10):761–767, 2004.
- [MHB<sup>+</sup>10] Elmar Mair, Gregory D. Hager, Darius Burschka, Michael Suppa, and Gerhard Hirzinger. Adaptive and generic corner detection based on the accelerated segment test. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 6312 LNCS, pages 183–196, 2010.
- [MM06] J. L. Marichal and M. J. Mossinghoff. Slices, slabs, and sections of the unit hypercube. *Online Journal of Analytic Combinatorics*, 2006.
- [MMK06] A Murarka, J Modayil, and B Kuipers. Building Local Safety Maps for a Wheelchair Robot using Vision and Lasers. In *Proceedings of the The 3rd Canadian Conference on Computer and Robot Vision*. IEEE Computer Society Washington, DC, USA, 2006.
- [MMM12] L. Moisan, P. Moulon, and P. Monasse. Automatic Homographic Registration of a Pair of Images, with A Contrario Elimination of Outliers. *IPOL*, 2:56–73, 2012.
- [MMM16] L. Moisan, P. Moulon, and P. Monasse. Fundamental Matrix of a Stereo Pair, with A Contrario Elimination of Outliers. *IPOL*, 6:89–113, 2016.
- [MMP15] D. Mishkin, J. Matas, and M. Perdoch. MODS: Fast and robust method for two-view matching. *CVIU*, 141:81–93, 2015.
- [MMRM17] Anastasiia Mishchuk, Dmytro Mishkin, Filip Radenovic, and Jiri Matas. Working hard to know your neighbor’s margins: Local descriptor learning loss. In *Advances in Neural Information Processing Systems*, pages 4826–4837, 2017.
- [MP04] P Moreels and P Perona. Common-frame model for object recognition. *Neural Information Processing Systems*, 2004.

- [MP05] P. Moreels and P. Perona. Evaluation of features detectors and descriptors based on 3d objects. In *ICCV*, pages 800–807, 2005.
- [MP07] P Moreels and P Perona. Evaluation of Features Detectors and Descriptors based on 3D Objects. *IJCV*, 73(3):263–284, 2007.
- [MRM18] D. Mishkin, F. Radenovic, and J. Matas. Repeatability is not enough: Learning affine regions via discriminability. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 284–300, 2018.
- [MS01] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *ICCV*, 1:525–531, 2001.
- [MS02] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. *ECCV*, 1:128–142, 2002.
- [MS04] K. Mikolajczyk and C. Schmid. Scale and Affine Invariant Interest Point Detectors. *IJCV*, 60(1):63–86, 2004.
- [MS05] K. Mikolajczyk and C. Schmid. A Performance Evaluation of Local Descriptors. *IEEE Trans. PAMI*, pages 1615–1630, 2005.
- [MSC<sup>+</sup>06] P. Musé, F. Sur, F. Cao, Y. Gousseau, and J.-M. Morel. An A Contrario Decision Method for Shape Element Recognition. *IJCV*, 69(3):295–315, 2006.
- [MSCG03] P. Musé, F. Sur, F. Cao, and Y. Gousseau. Unsupervised thresholds for shape matching. *ICIP*, 2003.
- [MTS<sup>+</sup>05] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A Comparison of Affine Region Detectors. *IJCV*, 65(1):43–72, 2005.
- [MY08] J M Morel and G Yu. On the consistency of the SIFT Method. Technical Report Prepublication, to appear in *Inverse Problems and Imaging (IPI)*, CMLA, ENS Cachan, 2008.
- [MY09] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.
- [MY11] J. M. Morel and G Yu. Is SIFT scale invariant? *Inv. Problems and Imaging*, 5(1):115–136, 2011.
- [NS06] D Nister and H Stewenius. Scalable recognition with a vocabulary tree. *CPVR*, pages 2161–2168, 2006.
- [NTG<sup>+</sup>06] A Negre, H Tran, N Gourier, D Hall, A Lux, and J L Crowley. Comparative study of People Detection in Surveillance Scenes. *Structural, Syntactic and Statistical Pattern Recognition, Proceedings Lecture Notes in Computer Science*, 4109:100–108, 2006.
- [OR15] E. Oyallon and J. Rabin. An Analysis of the SURF Method. *Image Processing On Line*, 5(2004):176–218, 2015. <https://doi.org/10.5201/ipol.2015.69>.

- [PH03] D Pritchard and W Heidrich. Cloth Motion Capture. *Computer Graphics Forum*, 22(3):263–271, 2003.
- [PLYP12] Y. Pang, W. Li, Y. Yuan, and J. Pan. Fully affine invariant SURF for image matching. *Neurocomputing*, 85:6–10, 2012.
- [Pĭ2] V. Pătrăucean. *Detection and Identification of Elliptical Structure Arrangements in Images: Theory and Algorithms*. PhD thesis, Institut National Polytechnique de Toulouse, France, 2012.
- [RAS18] I. Rocco, R. Arandjelovic, and J. Sivic. Convolutional neural network architecture for geometric matching. *TPAMI*, 2018.
- [RB16] Carolina Raposo and Joao P Barreto. Theory and practice of structure-from-motion using affine correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5470–5478, 2016.
- [RCP<sup>+</sup>13] R. Raguram, O. Chum, M. Pollefeys, J. Matas, and J-M. Frahm. USAC: a universal framework for random sample consensus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):2022–2038, 2013.
- [RDG09] J. Rabin, J. Delon, and Y. Gousseau. A statistical approach to the matching of local features. *SIIMS*, 2(3):931–958, 2009.
- [RDM18a] M. Rodriguez, J. Delon, and J.-M. Morel. Covering the space of tilts. application to affine invariant image comparison. *SIIMS*, 11(2):1230–1267, 2018.
- [RDM18b] M. Rodriguez, J. Delon, and J.-M. Morel. Fast affine invariant image matching. *IPOL*, 8:251–281, 2018.
- [RdSLD07] J Ruiz-del Solar, P Loncomilla, and C Devia. A New Approach for Fingerprint Verification Based on Wide Baseline Matching Using Local Interest Points and Descriptors. *Lecture Notes in Computer Science*, 4872:586, 2007.
- [RFGvG<sup>+</sup>19] M. Rodriguez, G. Facciolo, R. Grompone von Gioi, P. Musé, J.-M. Morel, and J. Delon. Sift-aid: boosting sift with an affine invariant descriptor based on convolutional neural networks. In *ICIP*, Sep 2019.
- [RFGvG<sup>+</sup>20] Mariano Rodriguez, Gabrielle Facciolo, Rafael Grompone von Gioi, Pablo Musé, and Julie Delon. Robust estimation of local affine maps and its applications to image matching. In *WACV*, 2020.
- [RFM<sup>+</sup>17] M. Rais, G. Facciolo, E. Meinhardt-Llopis, Morel J.-M., Buades A., and Coll B. Accurate motion estimation through random sample aggregated consensus. *CoRR*, abs/1701.05268, 2017.
- [RFvG<sup>+</sup>20] Mariano Rodríguez, Gabrielle Facciolo, Rafael Grompone von Gioi, Pablo Musé, Julie Delon, and Jean-Michel Morel. Cnn-assisted coverings in the space of tilts: best affine invariant performances with the speed of cnns. In *ICIP*, Oct 2020.
- [RGvG18] M. Rodriguez and R. Grompone von Gioi. Affine invariant image comparison under repetitive structures. In *ICIP*, pages 1203–1207, Oct 2018.

- [ROD14] I. Rey-Otero and M. Delbracio. Anatomy of the SIFT method. *IPOL*, 4:370–396, 2014.
- [RODM15] Ives Rey-Otero, Mauricio Delbracio, and Jean-Michel Morel. Comparing feature detectors: A bias in the repeatability criteria. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3024–3028. IEEE, 2015.
- [RTA06] F Riggi, M Toews, and T Arbel. Fundamental Matrix Estimation via TIP-Transfer of Invariant Parameters. *Proceedings of the 18th International Conference on Pattern Recognition (ICPR’06)-Volume 02*, pages 21–24, 2006.
- [SAS07] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In *MULTIMEDIA ’07: Proceedings of the 15th international conference on Multimedia*, pages 357–360, New York, NY, USA, 2007. ACM.
- [SI07] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *CVPR*, pages 1–8, 2007.
- [SLL01] S Se, D Lowe, and J Little. Vision-based mobile robot localization and mapping using scale-invariant features. *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, 2, 2001.
- [SSR<sup>+</sup>09] Cees Snoek, Kvd Sande, OD Rooij, Bouke Huurnink, J Uijlings, M van Liempt, M Bugalhoy, I Trancosoy, F Yan, M Tahir, et al. The MediaMill TRECVID 2009 semantic video search engine. In *TRECVID workshop*, 2009.
- [SSS06] N Snavely, S M Seitz, and R Szeliski. Photo tourism: exploring photo collections in 3D. *ACM Transactions on Graphics (TOG)*, 25(3):835–846, 2006.
- [SZ02] F Schaffalitzky and A Zisserman. Multi-view matching for unordered image sets, or How do I organize my holiday snaps? *ECCV*, 1:414–431, 2002.
- [SZ<sup>+</sup>03] Josef Sivic, Andrew Zisserman, et al. Video google: A text retrieval approach to object matching in videos. In *iccv*, volume 2, pages 1470–1477, 2003.
- [SZ14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [TV00] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. *BMVC*, pages 412–425, 2000.
- [TV04] T. Tuytelaars and L. Van Gool. Matching Widely Separated Views Based on Affine Invariant Regions. *IJCV*, 59(1):61–85, 2004.
- [TVO99] T. Tuytelaars, L. Van Gool, and Others. Content-based image retrieval based on local affinely invariant regions. *Int. Conf. on Visual Information Systems*, pages 493–500, 1999.

- [VGS10] Koen Van De Sande, Theo Gevers, and Cees Snoek. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1582–1596, 2010.
- [VL10] Christoffer Valgren and Achim J Lilienthal. SIFT, SURF & seasons: Appearance-based long-term localization in outdoor environments. *Robotics and Autonomous Systems*, 58(2):149–156, 2010.
- [VMU96] L. Van Gool, T. Moons, and D. Ungureanu. Affine/Photometric Invariants for Planar Intensity Patterns. *Proceedings of the 4th European Conference on Computer Vision-Volume I-Volume I*, pages 642–651, 1996.
- [VV05] M. Vergauwen and L. Van Gool. Web-based 3D Reconstruction Service. *Machine Vision and Applications*, 17(6):411–426, 2005.
- [VvHR05] M Veloso, F von Hundelshausen, and P E Rybski. Learning visual object definitions by observing human activities. In *Proc. of the IEEE-RAS Int. Conf. on Humanoid Robots*,, pages 148–153, 2005.
- [WRHS13] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1385–1392, 2013.
- [WT83] A. P. Witkin and J. M. Tenenbaum. On the role of structure in vision. In J. Beck, B. Hope, and A. Rosenfeld, editors, *Human and Machine Vision*, pages 481–543. Academic Press, 1983.
- [Yan07] K Yanai. Image collector III: a web image-gathering system with bag-of-keypoints. *Proc. of the 16th Int. Conf. on World Wide Web*, pages 1295–1296, 2007.
- [YC07] J Yao and W K Cham. Robust multi-view feature matching from multiple unordered views. *Pattern Recognition*, 40(11):3081–3099, 2007.
- [YM11] Guoshen Yu and Jean-Michel Morel. ASIFT: An Algorithm for Fully Affine Invariant Comparison. *IPOl*, 1:1–28, 2011.
- [YSST07] G Yang, C V Stewart, M Sofka, and C L Tsai. Alignment of challenging image pairs: Refinement and region growing starting from a single keypoint correspondence. *IEEE Trans. Pattern Anal. Machine Intell.*, 2007.
- [ZD06] Yefeng Zheng and David Doermann. Robust point matching for nonrigid shapes by preserving local neighborhood structures. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):643–649, 2006.
- [ZK15] Sergey Zagoruyko and Nikos Komodakis. Learning to compare image patches via convolutional neural networks. In *CVPR*, pages 4353–4361, 2015.
- [ZL16] Jure Zbontar and Yann LeCun. Stereo matching by training a convolutional neural network to compare image patches. *JMLR*, 17(1-32):2, 2016.
- [ZR11] C. L. Zitnick and K. Ramnath. Edge foci interest points. In *2011 International Conference on Computer Vision*, pages 359–366. IEEE, 2011.

- [ZYS09] Huiyu Zhou, Yuan Yuan, and Chunmei Shi. Object tracking using SIFT features and mean shift. *Computer vision and image understanding*, 113(3):345–352, 2009.